



Universitat Autònoma de Barcelona
Escola Tècnica Superior d'Enginyeria
Enginyeria Informàtica

SISTEMA DE CONTROL REMOT D'UNA MÀQUINA DE RAJOS X VIA WEB

Memòria del projecte de final de carrera corresponent als estudis d'Enginyeria Superior en Informàtica presentat per Mayte Mora Rodríguez i dirigit per David Castells i Rufas.

Bellaterra, Juny del 2007

El sotasignat, David Castells, professor de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha estat realitzada sota la seva direcció per Mayte Mora Rodríguez

Bellaterra, Juny del 2007

Signat: David Castells i Rufas

Resum

El projecte desenvolupat té com objectius la millora en la gestió de la configuració de les màquines de rajos X de l'empresa Varpe. La principal funció d'aquestes màquines és la inspecció d'envasos de productes alimentaris, amb l'objectiu de detectar anomalies produïdes durant l'envasat. En l'actualitat la resolució d'Incidències d'aquestes màquines requereix el desplaçament físic d'operaris a les localitzacions on estan situades les màquines. El projecte proporciona una nova forma de treball a partir de l'ús de bases de dades distribuïdes en les màquines de rajos X i que són gestionades per un servidor central. Entre d'altres, la principal avantatge d'aquesta gestió és que la reconfiguració i resolució d'incidències es pot realitzar fàcilment a través d'una aplicació web.

Resumen

El proyecto desarrollado tiene como objetivos una mejora en la gestión de la configuración de la máquinas de rayos X de la empresa Varpe. La principal función de estas máquinas es la inspección de los envases de productos alimentarios, con el objetivo de detectar anomalías producidas durante el envasado. En la actualidad la resolución de incidencias de estas máquinas requiere el desplazamiento físico de los operarios a las localizaciones donde están situadas las máquinas. El proyecto proporciona una nueva manera de trabajar a partir del uso de las bases de datos distribuidas en las máquinas de rayos X i que son gestionadas por un servidor central. Entre otras, la principal ventaja de esta gestión es que la reconfiguración i resolución de incidencias se puede realizar facilmente a través de una aplicación web.

Abstract

The main purpose of this work is to improve the configuration management of the X-ray machines installed by the company Varpe. The main function of these machines is to examine food containers looking for unusual objects. Currently, the configuration of such machines is carried out sending a specialized worker at the place where the device is located. The application developed in this work provides a new mechanism through the use of database distributed in the X-ray machines and managed by a central server. Among others, the main advantage of this application is that the device re-configuration and dealing with incidences can be easily carried out using a web application.

Agraïments

Vull aprofitar aquestes línies per agraïr el suport i paciència demostrada per David Castells i Rufas. Gràcies a ell aquest projecte a vist la llum i ha aconseguit els objectius marcats.

Donar les gràcies a LOS FABULOSOS per aquests anys on hem compartit pràctiques, riures, tristeses i moltes hores de biblioteca, gràcies a tots per fer el pas per la univeristat un camí de bons records.

Gràcies al departament d'energia d'AIA i a l'Alberto per cafés i experiències compartides durant la realització del projecte.

Gràcies als pares que m'han donat suport i ànims per tal de superar totes les etapes de la vida. Per últim però més importnat donar les gràcies al meu nino per aguantar el que ningú hagués aguantat, per fer-me riure cada dia durant els últims anys, per donar-me ànims en els moments més difícils i tranquil·litat en els moments de màxima ansietat.

A tots vosaltres, moltíssimes gràcies.

Mayte Mora Rodríguez

Índex

1	Presentació	1
1.1	Plantejament	1
1.2	Objectius	2
1.3	Estructura de la memòria	3
2	Anàlisis de requeriments	7
2.1	Estudi de l'aplicació	7
2.2	Requeriments de VARPE	8
2.3	Especificacions de la màquina de Rajos X	12
2.4	Eines utilitzades en el desenvolupament	13
2.4.1	La base de dades Paradox	14
2.4.2	La base de dades MySQL	14
2.4.3	El controlador JDBC-ODBC	16
2.4.4	<i>Java Server Pages</i> (JSP)	19
3	Disseny i desenvolupament	21
3.1	Disseny de l'aplicació	21
3.2	Sincronització de base de dades	21
3.2.1	Bases de dades replicades vs Bases de dades distribuïdes	22
3.2.2	Tècniques de sincronització de bases de dades	24
3.2.3	Especificacions del servidor	26
3.2.4	Mòdul de sincronització	27
3.3	Transmissió d'imatges	35

3.4	Aplicació Web	35
4	Validació de l'aplicació	41
4.1	Joc de proves	41
4.2	Resultats	44
5	Planificació	49
5.1	Planificació temporal	49
5.2	Anàlisi de costos	51
6	Conclusions	55
6.1	Resum	55
6.2	Futures millores	56
	Bibliografia	59
A	Algorisme d'inspecció	63

Capítol 1

Presentació

1.1 Plantejament

El projecte presentat a continuació soluciona un dels problemes amb els quals s'enfronta en l'actualitat l'empresa Varpe.

Varpe és una empresa catalana del sector de la metallúrgia que es dedica a la construcció de màquines de control de pes, màquines de rajos X i detectors de metalls. El projecte permetrà a Varpe un gestió global de les màquines de Rajos X. Aquestes màquines permeten visualitzar mitjançant rajos X el contingut de diferents productes com llet en pols, café soluble,... per tal de comprovar que a l'interior del producte no hi ha cap objecte estrany com un cargol, un clip, etc. Podríem dir que proporcionen una radiografia dels productes que passen per la màquina.

La màquina de rajos X Varpe permet analitzar uns 20 productes per minut traient de la cadena de producció aquells detectats com a contaminats, és a dir, aquells en que la màquina detecta que hi ha un objecte no desitjat.

Varpe distribueix aquestes màquines arreu del món. Les màquines de rajos X estan configurades mitjançant una paràmetres que permeten funcionar a la màquina de manera correcta. Aquests paràmetres s'emmagatzemen utilitzant una base de dades Paradox. Si aquests paràmetres no estan configurats la màquina pot eliminar de la cadena de producció productes que realment no estan contaminats.

Un dels principals problemes que actualment té Varpe és la dificultat que suposa solu-

cionar incidències de les màquines sense poder tenir accés a les dades de configuració, ja que aquestes són locals de la màquina.

Amb aquest projecte el que es proposa és tenir accés a aquestes dades via web i unificar les dades de totes les màquines de rajos X per tal de tenir-les emmagatzemades en un únic servidor, això si, identificant-les amb la màquina corresponent. D'aquesta manera, quan es produeix alguna incidència en alguna de les màquines els operaris només han d'executar el mòdul desenvolupat en el projecte per tal d'enviar la base de dades actualitzada que utilitza la màquina en el moment de la incidència. Després els empleats de Varpe a Barcelona poden comprovar que els paràmetres de configuració de la màquina siguin els correctes o modificar-los si fa falta.

1.2 Objectius

Els principals objectius del projecte eren proporcionar a l'empresa Varpe un mòdul de sincronització per replicar bases de dades i una aplicació web per visualitzar les dades. El mòdul de sincronització, permetrà al client de Varpe enviar les dades de la configuració de la màquina a un servidor central de Varpe. D'aquesta manera Varpe tindrà accés als paràmetres de configuració de la màquina d'aquell client.

L'aplicació web permetrà als operaris de Varpe visualitzar de manera agradable totes les dades existents en el servidor, d'aquesta manera podrà comprovar que els paràmetres de configuració de la màquina són els correctes.

Tot i que les parts esmentades són els objectius principals del projecte, aquest té altres objectius a assolir. Un dels objectius a assolir és permetre que el client de Varpe disposi d'una aplicació per tal d'enviar les imatges generades per la màquina al servidor de Varpe. Per últim caldrà integrar l'aplicació web implementada a la utilitzada per Varpe per tal de gestionar altres màquines com les de control de pes.

A continuació es troba un petit resum de tots els objectius que s'han d'assolir amb el desenvolupament del projecte.

- **Mòdul de sincronització:** Aquest és el principal objectiu a implementar en el projecte. El mòdul de sincronització es desenvoluparà en Java. Permetrà assolir un

dels requisits de Varpe, fer una réplica de les bases de dades que permeten configurar les màquines. Aquest mòdul ha de permetre crear de manera dinàmica totes les taules de la base de dades de la màquina de rajos X en el servidor MySQL. Un cop s'han creat les taules, el mateix mòdul permetrà replicar tota la base de dades Paradox en el servidor i actualitzar-les si ja existeixen.

- **Transferència d'imatges:** Un dels objectius del projecte és poder transferir les imatges que generen les màquines de Rajos X a un servidor i fer una web per tal de poder-les visualitzar. Aquest punt és important per tal de poder analitzar les imatges dels diferents productes generades per la màquina. Mitjançant l'anàlisi d'aquestes imatges l'operari de Varpe pot determinar si la màquina està detectant falsos positius degut a una configuració incorrecta.
- **Aplicació Web:** El darrer objectiu del projecte és la creació d'una sèrie de pàgines web per tal de poder visualitzar totes les dades del servidor de manera amigable.
- **Integració:** Un cop desenvolupat el projecte, Varpe vol que aquest s'integri amb l'aplicació Web que tenen en l'actualitat. Varpe gestiona les seves màquines associant-les a un client i una fàbrica. Es per aquesta raó que el disseny de les taules del nou servidor ha de permetre una fàcil integració tenint en compte els possibles conflictes de duplicat de claus o problemes d'integritat referencial.

1.3 Estructura de la memòria

La memòria està estructurada en sis capítols. El capítol 1 està destinat a fer una presentació del projecte. En el segon capítol es fa un estudi dels requeriments i es presenta una petita introducció de les eines utilitzades pel desenvolupament del projecte. El tercer capítol és el capítol principal on s'explica la implementació del projecte. El capítol quart fa referència a les diferents proves i resultats realitzats amb la implementació proposada pel projecte. Al capítol cinc es fa un anàlisi sobre els costos del projecte i es troba la planificació per dur a terme el projecte. Finalment el capítol de les conclusions conclou aquesta memòria.

En termes generals, cada capítol està estructurat de la següent forma:

- El capítol 2 titulat “Anàlisi de requeriments” està format pels següents apartats:
 - **Estudi de l’aplicació:** En aquest apartat podem trobar una visió general de les diferents parts que componen el projecte.
 - **Requeriments de Varpe:** Explicació dels requeriments exposats per l’empresa Varpe per desenvolupar el projecte.
 - **Especificacions de la màquina de Rajos X:** En aquest apartat trobem una breu explicació del funcionament de la màquina de Rajos X.
 - **Eines utilitzades en el desenvolupament:** Explicació de les eines necessàries pel desenvolupament del projecte.
 - * La base de dades Paradox: Introducció a les bases de dades Paradox.
 - * La base de dades MySQL: Introducció a les bases de dades MySQL.
 - * El controlador JDBC-ODBC: Explicació del controlador utilitzat per implementar el mòdul de sincronització.
 - * Java Server Pages (JSP): Explicació de l’eina utilitzada per desenvolupar l’aplicació web.
- El capítol 3 titulat “Disseny i desenvolupament” està compost pels següents apartats:
 - **Disseny de l’aplicació:** En aquest apartat s’explica el disseny general del projecte desenvolupat.
 - **Sincronització de bases de dades:** Aquest és el mòdul principal de projecte en ell trobem l’explicació de com s’ha realitzat el mòdul de sincronització.
 - * Bases de dades replicades vs Bases de dades distribuïdes: En aquest apartat s’explica les principals diferències entre tenir la réplica d’una base de dades i tenir una base de dades distribuïda.
 - * Tècniques de sincronització de bases de dades: En aquest apartat s’expliquen les diferents opcions que es disposen per replicar bases de dades.
 - * Especificacions del servidor: En aquest apartat s’expliquen les possibles maneres d’administrar el servidor Varpe.

- * Mòdul sincronització: En aquest apartat es trobem els algorismes implementats en el mòdul de sincronització.
- **Transmissió d’imatges:** Apartat on s’explica la manera utilitzada per enviar imatges des de la màquina fins el servidor Varpe.
- **Aplicació Web:** En aquest apartat s’explica el funcionament de l’aplicació web desenvolupada per tal de gestionar les dades del servidor.
- El capítol 4 titulat “Validació de l’aplicació” consta de dos apartats:
 - **Joc de proves:** En aquest apartat trobem les diferents proves realitzades per validar en eficiència i eficàcia l’aplicació.
 - **Resultats:** Anàlisi dels resultats obtinguts en realitzar el joc de proves.
- El capítol 5 titulat “Planificació” presenta la planificació proposada per tal de desenvolupar el projecte. Aquest capítol està dividit en dos seccions:
 - **Planificació temporal:** En aquesta secció es pot veure la primera planificació feta per portar a terme el projecte.
 - **Anàlisi de costos:** Explicació dels costos associats a l’elaboració del projecte.
- Per últim tenim el capítol de Conclusions.
 - **Resum:** Resum final sobre l’elaboració del projecte i objectius aconseguits.
 - **Futures millores:** Proposta de millores per tal d’aconseguir un millor rendiment del mòdul de sincronització.

Capítol 2

Anàlisi de requeriments

Aquest capítol es centra en l'explicació general del projecte realitzat. En aquest capítol es pot trobar els diferents requeriments proposats per Varpe del que hauria de ser el projecte per tal de satisfer les seves necessitats actuals.

Un cop analitzats els requeriments s'expliquen les eines utilitzades per tal de desenvolupar el projecte.

2.1 Estudi de l'aplicació

En primer pas Varpe va proporcionar les taules que podem veure a la Figura 2.1. Les taules proporcionades primerament en Paradox i a mesura que va avançar el projecte en MySQL no tenen cap mena d'integritat referencial. Tot i compondre una base de dades no hi ha cap relació establerta entre elles. Una altra de les incoherències en les taules proporcionades és la manca de clau primària per tal de definir de manera única els registres de les taules. Amb el disseny de taules proporcionat es podia donar el cas de tenir registres duplicats. Per tal de solucionar aquest problema d'integritat les taules creades en el servidor MySQL segueixen l'esquema de la Figura 2.2. Les claus primàries escollides són la IP de la màquina, ja que la IP identifica de manera única una màquina i el primer camp de la taula. Aquest camp en la majoria dels casos són la ID del registre així que la combinació dels dos camps identificarà de manera única el registre.

Tot i que en el projecte no s'ha implementat la integritat referencial de les taules ja

config	
Configuration	Int(11)
DetX	double
Lines	double
MaxKv	double
MaxmA	double
ScreenX	double
ScreenY	double
Fmask	double
Lanes	double
ProductFile	varchar(40)
Gens	double
DetCOM	double
XgenCOM	double
Maximages	double

Products	
Product_ID	Int(11)
Station_ID	Int(11)
Product_Name	Varchar(40)
A10	Int(11)
....	
A40	Int(11)

Parame	
Param	Int(11)
ParamName	Varchar(40)
ParamGroup	Int(11)
ALevel	Int(11)
Mi_	Int(11)
Ma_	Int(11)

Interlock	
Illno	Double
Name	Varchar(40)
Bi_	Double
Sense	Double
Help	Varchar(40)

Param	
Param	Int(11)
ParamName	Varchar(40)
ParamGroup	Int(11)
ALevel	Int(11)
Mi_	Int(11)
Ma_	Int(11)

log	
Log_ID	Int(11)
Dat_	Varchar(40)
Tim_	Varchar(40)
Message	Varchar(40)

alignment	
Inde_	double
Align Kv	double
Align mA	double
Run Kv	double
Run mA	double
Cal_ Kv	double
Cal_ mA	double
Scan Rate	double

groupe	
Grou_	Int(11)
Group Name	Varchar(40)
Special	Int(11)

PWD	
Name	varchar(40)
Level_	Double
Password	double
Language_	Char(1)



group	
Grou_	Int(11)
Group Name	Varchar(40)
Special	Int(11)



Figura 2.1: Taules proporcionades per Varpe



que un dels requeriments de Varpe és que les seves taules no poden ser modificades i que el servidor ha de tenir una rèplica de les mateixes, s'ha implementat de manera teòrica el que podria ser el diagrama entitat-relació de la base de dades (vegeu Figura 2.3)



2.2 Requeriments de VARPE



En l'actualitat l'empresa Varpe, disposa de diferents màquines de rajos X. Aquestes màquines són configurades mitjançant una base de dades Paradox amb diferents taules on es guarda la informació que fa referència a la llibreria de productes analitzats per la màquina i els paràmetres que permet configurar la màquina per tal que el reconeixement que fa



config	
Configuration 	Int(11)
Maquina 	varchar(20)
DetX	double
Lines	double
MaxKv	double
MaxmA	double
ScreenX	double
ScreenY	double
Fmask	double
Lanes	double
ProductFile	varchar(40)
Gens	double
DetCOM	double
XgenCOM	double
Maximages	double



log	
Log_ID 	Int(11)
Maquina 	varchar(20)
Dat_	Varchar(40)
Tim_	Varchar(40)
Message	Varchar(40)



PWD	
Name 	varchar(40)
Maquina 	varchar(20)
Level_	Double
Password	double
Language_	Chax(1)



Products	
Product_ID 	Int(11)
Maquina 	varchar(20)
Station_ID	Int(11)
Product_Name	Varchar(40)
A10	Int(11)
....	
A40	Int(11)



alignment	
Inde_ 	double
Maquina 	varchar(20)
Align Kv	double
Align mA	double
Run Kv	double
Run mA	double
Cal_ Kv	double
Cal_ mA	double
Scan Rate	double

Interlock	
Illno 	Double
Maquina 	varchar(20)
Name	Varchar(40)
Bi_	Double
Sense	Double
Help	Varchar(40)

Images	
ID 	Int(11)
Maquina 	varchar(20)
Name	Varchar(40)
Photo	longblob

Param	
Param 	Int(11)
Maquina 	varchar(20)
ParamName	Varchar(40)
ParamGroup	Int(11)
ALevel	Int(11)
Mi_	Int(11)
Ma_	Int(11)

Parame	
Param 	Int(11)
Maquina 	varchar(20)
ParamName	Varchar(40)
ParamGroup	Int(11)
ALevel	Int(11)
Mi_	Int(11)
Ma_	Int(11)

groupe	
Grou_ 	Int(11)
Maquina 	varchar(20)
Group Name	Varchar(40)
Special	Int(11)



group	
Grou_ 	Int(11)
Maquina 	varchar(20)
Group Name	Varchar(40)
Special	Int(11)

Figura 2.2: Taules creades en el servidor

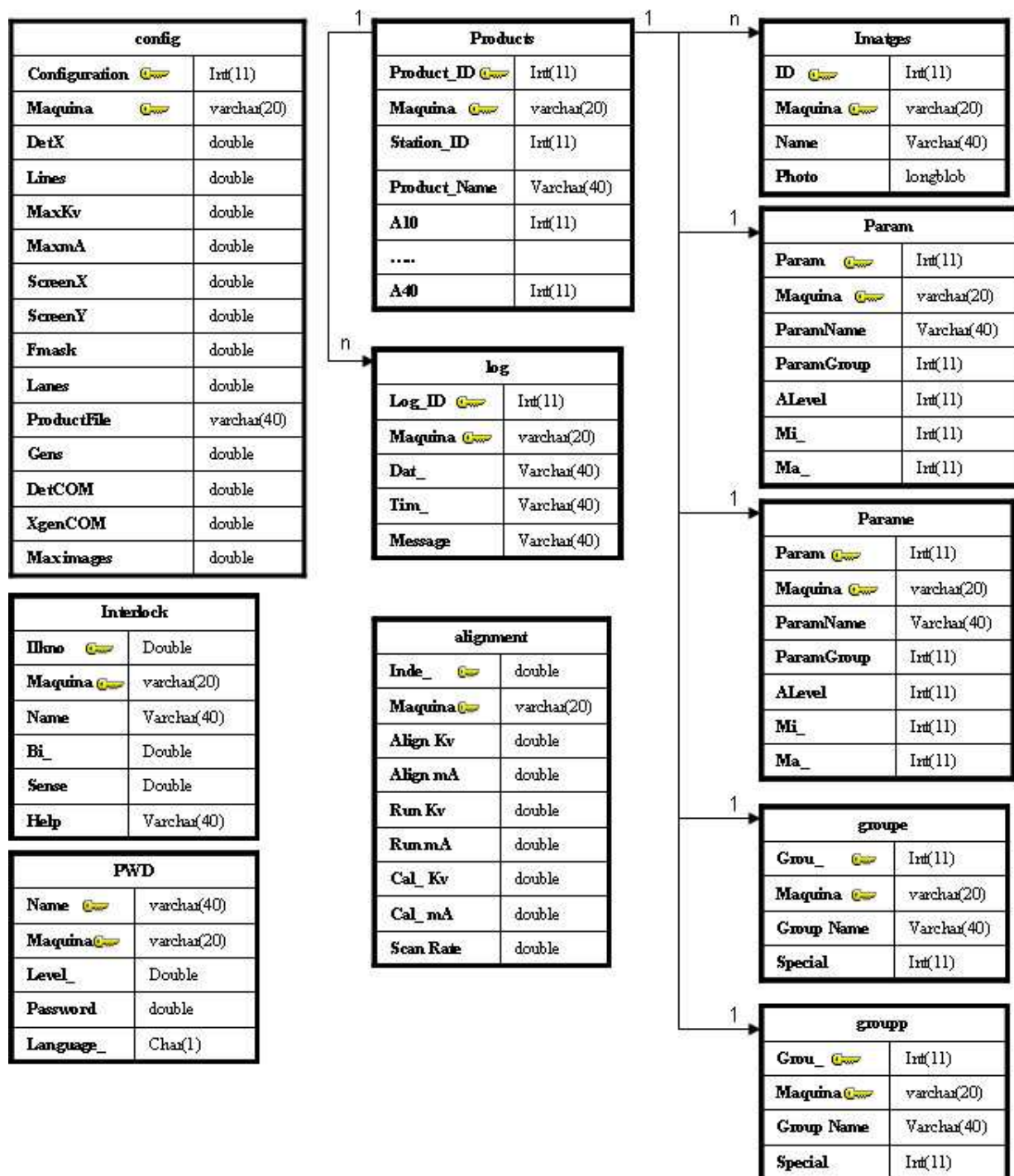


Figura 2.3: Diagrama entitat relació proposat

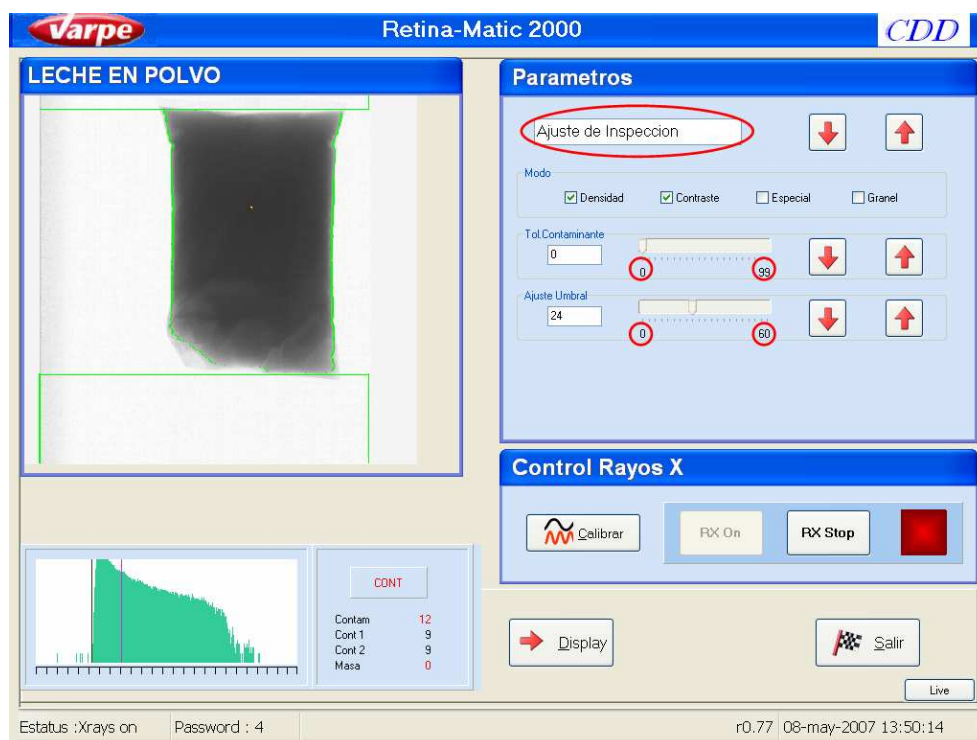
sigui correcte i no generi falsos positius. Les taules que componen la base de dades són les següents:

- Products: Taula on es registren el productes analitzats per la màquina.
- Alignment: Taula on s'especifica l'alineament de la configuració.
- Config: Taula on es guarda la configuració específica de cada producte.
- Log: Taula on es registren els diferents logs generats per la màquina.
- Interlock: Taula on es guarden noms per defecte.
- pwd: Taula que permet gestionar privilegis i password dels usuaris de la màquina.
- paramX: Taula on es guarden els paràmetres que configuren cada producte.
- groupX: Taula que gestiona les cadenes de caràcters que apareixen en el display de la màquina.

La X fa referència a l'idioma en el qual estan guardades les dades de les taules. Per exemple paramp vol dir que les dades estan guardades en portuguès i groupe vol dir que les dades estan guardades en anglès.

El problema de l'estructura muntada per VARPE és que la base de dades que permet configurar la màquina només era possible consultar-la de manera local. Això fa que en l'actualitat si un operari de Varpe vol comprovar que els paràmetres de la màquina són els correctes s'ha de desplaçar fins on està la màquina. Aquest fet suposa un cost extra per Varpe és per aquesta raó que l'empresa proposa les següents millores en les seves aplicacions:

- Es vol centralitzar totes les dades generades per les diferents màquines en una única base de dades, en aquest cas MySQL
- Un requeriment important per Varpe, és que les bases proporcionades no poden ser modificades, ja que tenen un format determinat per tal de configurar de manera correcta la màquina de rajos X.

Figura 2.4: *Interfície màquina rajos X*

- Es vol tenir una aplicació web que permeti veure les dades del servidor MySQL.
- La màquina de Rajos X pot guardar una sèrie d'imatges que Varpe també vol poder visualitzar via web.
- Per últim es proposa fer la integració del projecte dins la web actual de Varpe.

2.3 Especificacions de la màquina de Rajos X

El funcionament de la màquina de Rajos X de Varpe és semblant a les utilitzades pels metges en els seus consultoris. A l'interior de la màquina es troba un tub de rajos X, dins del tub es troba un canó que dispara electrons molt energitzats. Els rajos X es generen mitjançant un procés atòmic induït pels electrons energitzats. Els rajos X són com qualsevol radiació electromagnètica i poden ser produïts en paquets d'energia anomenats fotons de la mateixa manera que la llum.

Un cop tenim en funcionament la màquina en la pantalla tàctil de la màquina apareix una pantalla semblant al de la Figura 2.4. Abans de posar en funcionament la màquina cal calibrar-la i escollir el producte que volem analitzar de la llibreria de productes que

disposa la màquina. Aquesta pantalla també permet a l'usuari, depenent dels privilegis atorgats, crear un nou producte en la llibreria. La llista del productes disponibles per la màquina són carregats a través de la taula products de la base de dades(vegeu Figura 2.3), un cop seleccionat el producte la màquina carrega els paràmetres (taula param) associats al producte. Aquests paràmetres recuperats poden ser modificats per l'usuari mitjançant la pantalla tàctil de que disposa la màquina.

Els registres emmarcats amb el cercle vermell de la Figura 2.4 són determinats per la taula param de la base de dades. Aquests paràmetres són específics de cada producte i permeten que l'anàlisi del producte sigui el correcte.

Un cop tenim totes les dades carregades a la màquina introduïm el producte que es vol examinar per la cinta transportadora de la màquina. En el moment que el producte passa per la màquina mitjançant l'algorisme d'inspecció es recupera la imatge del producte examinat. A l'apèndix A es troba l'explicació de l'algorisme utilitzat per Varpe per tal de fer la reconstrucció de la imatge del producte examinat, aquest algorisme ha de permetre una ràpida reconstrucció de la imatge, ja que en el moment de sortida del producte l'usuari a de poder visualitzar la imatge a la pantalla de la màquina per poder determinar si el productes està contaminat o no.

Si s'ha trobat algun objecte estrany, la màquina determina que aquell producte està contaminat i el treu de la cadena de producció. A la Figura 2.5 podem veure la imatge proporcionada per la màquina d'un producte contaminat, la part groga pertany a l'objecte estrany, en aquest cas un clip d'oficina.

Pel tal que la màquina funcioni correctament i sigui capaç de determinar quins productes estan contaminats i quins no, és molt important que els paràmetres associats al producte estiguin configurats de manera correcta. En cas contrari podrien tenir falsos positius i la màquina eliminaria de la cadena de producció productes no contaminats.

2.4 Eines utilitzades en el desenvolupament

En aquesta secció s'expliquen les principals eines que s'han utilitzat per tal d'implementar el projecte. El projecte ha estat desenvolupat en dues fases: la primera fase centrada en

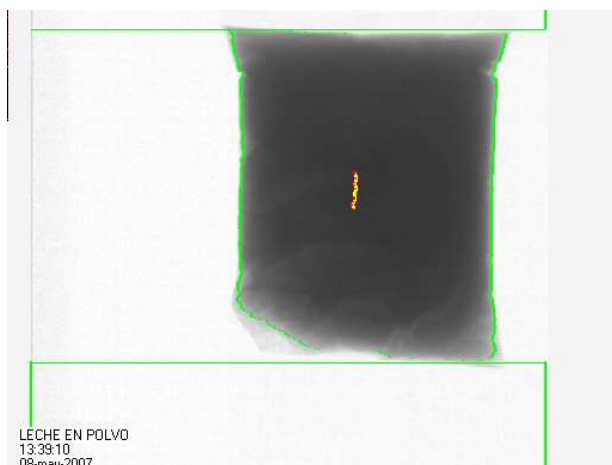


Figura 2.5: *Producte contaminat*

la base de dades Paradox i la segona basada en la migració a MySQL. És per aquesta raó que les principals eines utilitzades, i que s'expliquen a continuació, tenen a veure tant amb Paradox com amb MySQL.

2.4.1 La base de dades Paradox

Paradox és una base de dades relacional per entorn Windows. En els seus inicis estava disponible per MS-DOS i Linux. Actualment està desenvolupada per Corel i és afegida en el pack de WordPerfect Office.

Al inici del projecte les bases de dades que proporcionaven les màquines de Rajos X per les quals s'ha desenvolupat el projecte eren d'aquest tipus.

En l'actualitat aquest tipus de base de dades no s'utilitza és per aquesta raó que van sorgir problemes amb els drivers per Paradox. Existeixen diferents aplicacions que permeten passar.

2.4.2 La base de dades MySQL

MySQL és un sistema de gestió de base de dades sota la llicència GPL (*General Public License*). GPL és un tipus de llicència creada per la *Free Software Foundation* que permet la lliure distribució i ús de software.

Primerament MySQL va ser desenvolupat per l'empresa opensource MySQL AB, fundada per David Axmark, Allan Larsson i Michael Widenius. MySQL pretén complir amb

l'estàndard SQL sense perdre velocitat, fiabilitat i portabilitat.

En l'actualitat MySQL és molt utilitzat en aplicacions web, la seva popularitat està relacionada amb PHP (*PHP Hypertext Pre-processor*). MySQL és una base de dades molt ràpida en el que fa referència a la lectura quan utilitza el model transaccional MyISAM però pot ocasionar problemes relacionats amb la integritat referencial.

MySQL suporta diferents tipus de taules. Els tipus de taules utilitzades en el desenvolupament del projecte són les taules MyISAM i les taules InnoDB. Les taules InnoDB estan estructurades de manera diferent que les MyISAM ja que aquestes estan guardades en tres arxius, en canvi les taules InnoDB es guarden en un únic arxiu.

Les taules de tipus InnoDB permeten definir regles d'integritat referencial i treballar amb transaccions. El fet de poder treballar amb transaccions aporta fiabilitat a la base de dades. Aquest tipus de taules aporten seguretat a les operacions tipus UPDATE, ens asseguren que les operacions s'han realitzat amb totalitat, és a dir que si al realitzar l'UPDATE succeïx algun esdeveniment pel qual aquest no pot arribar al final de l'execució el registre es queda en el seu estat inicial. És per aquesta raó que és recomanable utilitzar aquest tipus de taules quan tenim un tipus d'estructura client servidor, ja que si estem executant una transacció i es perd la connexió entre el servidor i el client no es perd la integritat de les dades. Un dels inconvenients d'aquest tipus de taules és el temps que triguen les transaccions en executar-se.

Per altra banda tenim les taules MyISAM tot i que aquestes s'estan substituint per les taules InnoDB. Aquest tipus de taules és caracteritzen per la gran velocitat que obtenen en les consultes ja que no han de fer comprovacions per tal de mantenir la integritat referencial. Aquest tipus de taules són utilitzades en aplicacions web on no existeixen un gran número de insercions en la base de dades.

A la Figura 2.6 podem veure com es crea una taula InnoDB en MySQL. És important recordar que quan no s'especifica el tipus de la taula al crear-la, MySQL determina que aquesta és de tipus MyISAM.

```

MySQL Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 5.0.27-community-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use varpe;
Database changed
mysql> CREATE TABLE `alignmentInnoDB` (
  -> `Inde_` double default NULL,
  -> `Align Kv` double default NULL,
  -> `Align mA` double default NULL,
  -> `Run Kv` double default NULL,
  -> `Run mA` double default NULL,
  -> `Cal_ Kv` double default NULL,
  -> `Cal_ mA` double default NULL,
  -> `Scan rate` double default NULL
  -> ) ENGINE=InnoDB;
Query OK, 0 rows affected (0.11 sec)

mysql>

```

Figura 2.6: Creació de taules InnoDB.

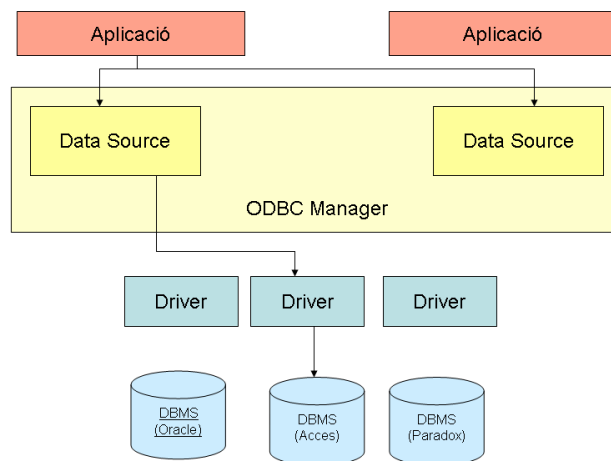


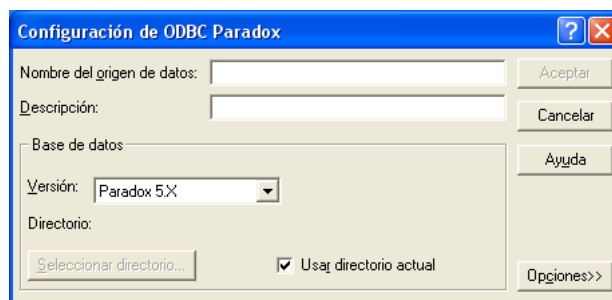
Figura 2.7: Estructura del driver ODBC

2.4.3 El controlador JDBC-ODBC

Per tal de poder tractar les dades Paradox en Windows, sense necessitat d'instal·lar Paradox, és necessari instal·lar el controlador ODBC (*Open DataBase Connectivity*). Aquest driver desenvolupat per Microsoft ens permet accedir a les dades de qualsevol base de dades sense que importi el DBMS (*Database Management System*), en aquest cas Paradox.

A la Figura 2.7 es pot observar com l'aplicació s'encarrega de fer la crida al controlador ODBC i aquest carrega el *ODBC Manager*. El *ODBC Manager* s'encarrega de determinar el driver corresponent i estableix la connexió amb el DBMS corresponent.

El controlador ODBC proporciona una capa que ens permet fer una traducció de les comandes generades per l'aplicació a comandes que el DBMS pugui entendre. Cal tenir

Figura 2.8: *Interfície Windows.*

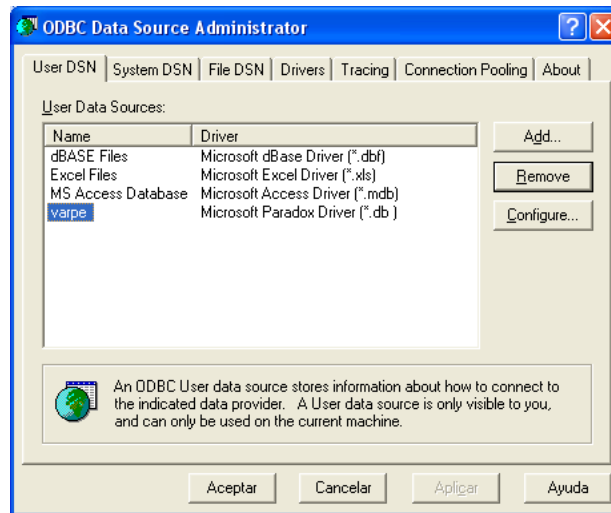
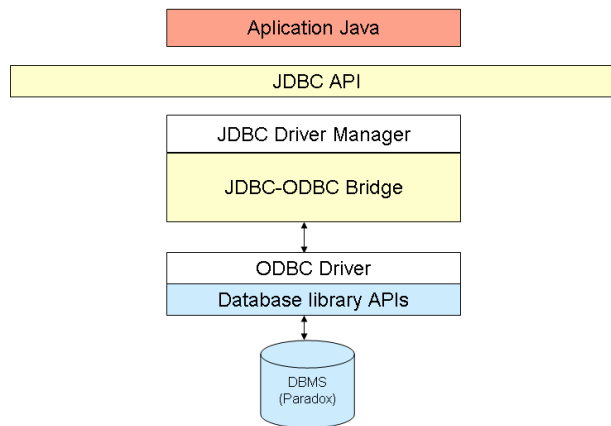
present que per poder fer aquesta traducció tant l'aplicació com el DBMS siguin compatibles amb el controlador ODBC. En aquest punt és on van aparèixer els problemes amb la base de dades Paradox ja que la versió en que van ser proporcionades les taules no eren compatibles amb el controlador ODBC que disposàvem. Aquest és un dels principals motius pels quals Varpe comença la migració de la seva base de dades Paradox a MySQL.

Per tal d'establir una connexió ODBC per poder comunicar l'aplicació i la base de dades cal definir un *Data Source Name*(DNS). El DNS ens permet assignar el nom desitjat als paràmetres necessaris per tal de connectar amb una base de dades ODBC concreta. Es pot considerar el DNS com un tipus d'accés directe que es crea en Windows per representar una connexió.

A Figura 2.8 podem veure la interfície que ens proporciona Windows per tal d'escollir la base de dades a la qual volem assignar el controlador ODBC que en aquest cas és ODBC-Paradox perquè recordem que la base de dades proporcionada per Varpe és Paradox.

Un cop escollida la base de dades podem veure a la Figura 2.9 com el controlador queda enregistrat en el *ODBC Data Source Administration*. Així doncs, seguint l'esquema de la Figura 2.7 i un cop configurat el controlador ODBC si per exemple es volgués obrir la bases de dades Paradox mitjançant l'Access es carregaria el driver Varpe, que com podem veure a la Figura 2.9 és un *Microsoft Paradox driver*.

En el moment en que s'ha associat la base de dades al controlador ODBC cal establir una relació entre el controlador ODBC i el controlador *Java Database Connectivity*(JDBC) mitjançant un connector proporcionat per Sun Microsystems tal i com podem veure a la Figura 2.10. Aquest pas és necessari degut a que el mòdul implementat, que necessita fer un ús de les dades Paradox, s'ha desenvolupat en Java. Per tal de fer aquesta associació

Figura 2.9: *ODBC Data Source Administration.*Figura 2.10: *Estructura JDBC-ODBC.*

només cal instal·lar el controlador pont JDBC-ODBC [6].

El controlador JDBC ens permet executar diferents operacions sobre la base de dades utilitzant Java. La *Application Program Interface*(API) JDBC està composta per diferents interfícies Java i mètodes de gestió. Aquesta interfície ens permet fer consultes a la base de dades utilitzant comandes *Structured Query Language*(SQL). En l'actualitat existeixen dos nivells d'interfícies (Figura 2.10), la interfície principal i l'administrador JDBC que és el que es comunica amb els controladors.

Per tal de realitzar operacions de consulta, actualitzacions,... sobre la base de dades cal establir una connexió. La manera d'establir aquesta connexió s'explica en el punt 3.2.4.

Un cop feta la migració a MySQL de les taules Paradox el procés explicat anteriorment no és necessari. Sun Microsystems proporciona un pont i un driver que permeten accedir

directament a les dades de la base de dades MySQL sense necessitat de passar el JDBC a ODBC.

2.4.4 *Java Server Pages*(JSP)

Java Server Pages és una tecnologia que permet crear aplicacions web. Va ser desenvolupat per Sun Microsystems i el seu funcionament es basa en el llenguatge Java. Aquesta tecnologia permet generar contingut dinàmic en pàgines web.

Les avantatges d'aquest tipus de tecnologia són els següents:

- Es podria dir que la tecnologia JSP és la competència de la tecnologia ASP(*Active Server Pages*) desenvolupada per Microsoft. Una de les principals avantatges de JSP enfront ASP és la portabilitat que ofereix el poder programar la part dinàmica de la web en Java.
- Les pàgines JSP poden contenir codi HTML/XML barrejant etiquetes especials per tal de programar scripts executables en el servidor en sintaxis Java.
- Si comparem JSP amb HTML aquest últim no pot contenir informació dinàmica. Aquest fet fa que les pàgines en HTML no puguin estar basades en fonts de dades contingudes en un servidor.

Per tal d'utilitzar aquesta tecnologia cal tenir un servidor Web que capaç de suportar pàgines HTML i JSP. En el cas del projecte el servidor escollit ha estat Tomcat[10]. Abans d'executar les pàgines .jsp el servidor web les compila i genera un servlet. Els servlet són objectes executats dins el context d'un contenidor de servlets com és Tomcat.

Capítol 3

Disseny i desenvolupament

En aquest capítol es troba l'explicació detallada de la implementació del projecte utilitzant diferents esquemes explicatius.

3.1 Disseny de l'aplicació

A la Figura 3.1 es pot veure un esquema general del projecte desenvolupat. En termes generals una base de dades replicada en un servidor i visualització de les dades mitjançant l'aplicació web.

L'aplicació dissenyada en el projecte es centra en el mòdul de sincronització. Mitjançant aquest mòdul es permet al client de Varpe crear l'estructura de les seves taules en el servidor MySQL de Varpe, fer la rèplica de tots els registres que té la base de dades i en algun moment donat fer una actualització d'aquests registres.

Un cop el servidor MySQL conté els registres de les diferents màquines per tal de visualitzar les dades s'ha creat una aplicació web.

3.2 Sincronització de base de dades

El principal mòdul desenvolupat en el projecte és el mòdul de sincronització. Aquest mòdul permet crear la base de dades en el servidor on seran guardades totes les dades de les diferents màquines.

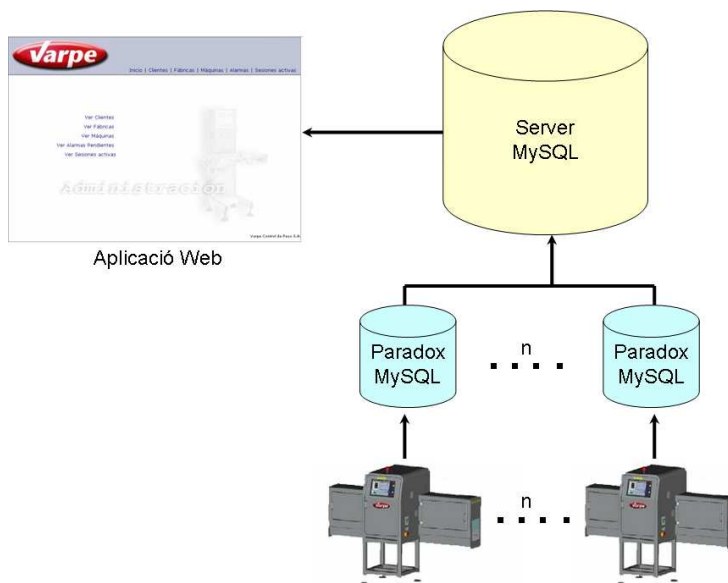


Figura 3.1: *Esquema de l'aplicació*

El mòdul desenvolupat en el projecte permet passar les bases de dades Paradox de les diferents màquines de Rajos X a un servidor MySQL.

Per tal de desenvolupar el mòdul s'han utilitzat els paquets `java.sql` i `java.sql` proporcionats per Sun Microsystems i que ens permeten accedir a les bases de dades mitjançant comandes SQL. Cal tenir present que per unificar totes les bases de dades de les diferents màquines en una sola s'ha hagut d'afegir un camp en el server per tal d'identificar la màquina a la qual pertany el registre.

3.2.1 Bases de dades replicades vs Bases de dades distribuïdes

Existeix una gran diferència entre les bases de dades replicades i les bases de dades distribuïdes. Una base de dades replicada té exactament les mateixes dades que la base de dades de la qual s'ha fet la rèplica. Pel contrari, en una base de dades distribuïda no tenim totes les dades disponibles en un servidor central sinó que les dades estan distribuïdes en diversos servidors.

Per rèplica s'entén el procés de copiar i mantenir les relacions en diferents bases de dades que formen un sistema distribuït. Tot i que parlem de rèplica el servidor MySQL no tindrà exactament els mateixos registres que la base de dades de la màquina, ja que per tal de diferenciar quin registre pertany a cada màquina s'ha hagut d'afegir un nou camp

on guardar la IP de la màquina.

La rèplica de la base de dades de les màquines al servidor de Varpe, permetrà als operaris tenir accés a les dades de configuració de cadascuna de les màquines.

Els avantatges que suposa la rèplica d'una base de dades són els següents:

- **Disponibilitat:** La rèplica d'una base de dades incrementa la disponibilitat de les dades pels usuaris.
- **Fiabilitat:** El terme fiabilitat fa referència al fet que al tenir guardades les mateixes dades en servidors diferents proporciona un mecanisme de recuperació de dades en calent en el cas que es produeixi algun tipus d'error.
- **Rendiment:** Proporciona un accés ràpid i local a les dades replicades ja que uns usuaris poden accedir a un servidor i altres a la rèplica d'aquest. Cal tenir present que per aportar aquesta avantatge cal que les rèpliques estiguin actualitzades en tot moment.
- **Reducció de les transaccions:** Les transaccions es poden dividir entre els servidors que contenen la mateixa informació d'aquesta manera es redueix el volum de transaccions que ha de processar el servidor.
- **Multi-usuari:** Les rèpliques de bases de dades permeten crear diferents instàncies (snapshots) personalitzades que satisfan els requisits de cada usuari del sistema.

Tot i que en el cas del projecte desenvolupat està clar que s'ha de fer una rèplica ja que Varpe només té un servidor central el sistema de bases de dades distribuïdes seria una bona opció si Varpe fos una multinacional. D'aquesta manera a cada servidor tindríem la informació en l'idioma corresponent i evitarien tenir taules a la base de dades on només canvia l'idioma dels camps de tipus varchar.

Les avantatges que proporciona un sistema distribuït són:

- **Utilització compartida de les dades:** Aquesta és una de les principals avantatges d'aquest tipus de sistema ja que permet un control de les dades de manera local.

- **Fiabilitat i disponibilitat:** Si es produeix un error en un dels servidors del sistema distribuït els usuaris que treballen amb les dades guardades en els altres servidors poden seguir treballant. La disponibilitat és important en sistemes utilitzats en aplicacions de temps real.
- **Velocitat:** Si una consulta necessita dades de diferents servidors aquesta es pot dividir en subconsultes i executar-se de manera paral·lela.

3.2.2 Tècniques de sincronització de bases de dades

Com que l'opció escollida en el projecte és la rèplica d'una base de dades, en fer una rèplica una de les característiques més importants que ha de complir la rèplica és que aquesta estigui sincronitzada amb la base de dades corresponent.

Existeixen diferents tècniques per tal de replicar una base de dades. A continuació s'expliquen les diferents tècniques i quina ha estat implementada en el projecte. Cal recordar que al no tenir privilegis per canviar les taules generades per les màquines, ja que aquestes taules exceptuant la taula imatges han estat proporcionades per l'empresa VARPE, alguna de les tècniques explicades a continuació no ha pogut ser finalment implementada.

Existeixen dos tipus de replicació la replicació síncrona i la replicació asíncrona.

- **Replicació Síncrona:** En aquest tipus de replicació les dades es repliquen en el moment en que han estat actualitzades en el client. Normalment utilitzen el protocol *Two Phase Commit*(2PC). Aquest protocol permet controlar l'execució d'una transacció distribuïda. El protocol està dividit en dues fases, d'aquí el seu nom, la fase de petició i la fase de commit[9] (vegeu Figura 3.2). Un dels principals inconvenients del protocol és que és un protocol bloquejant. Per tant, els processos compateixen entre ells per tal de bloquejar el recurs.

Aquest tipus de replicació és perfecte per estructures client-servidor on el servidor hagi d'estar sempre sincronitzat amb els seus clients. Un dels principals inconvenients que té aquest sistema de replica és que si el servidor on s'ha de fer la replica no està disponible la transacció fallarà en el client.

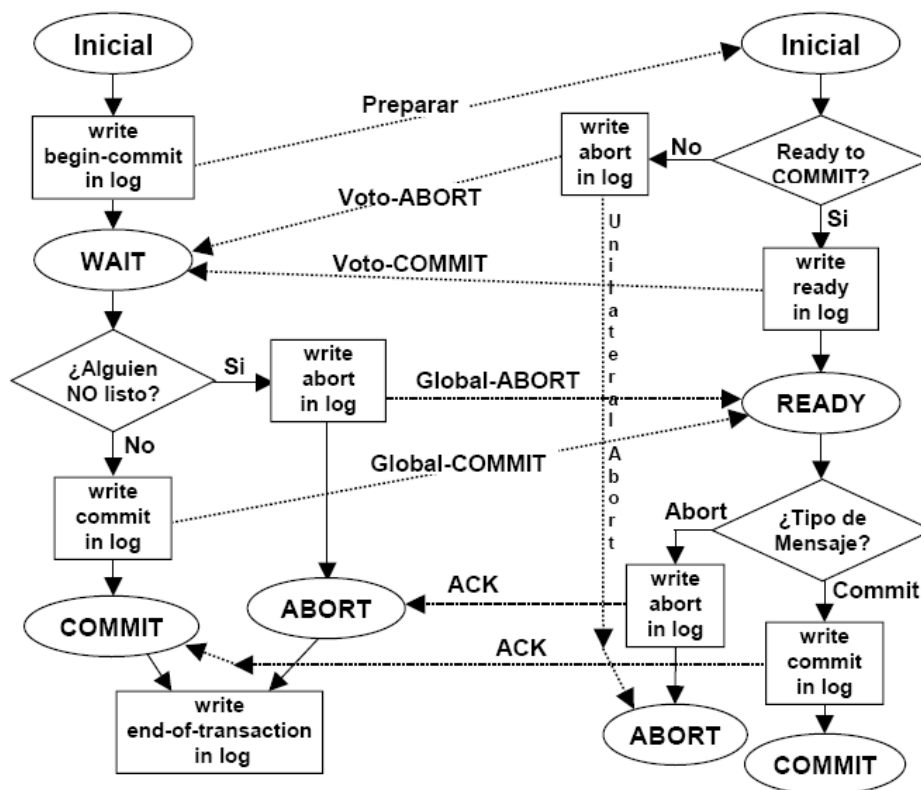


Figura 3.2: Accions del protocol 2PC

- **Replicació Asíncrona:** En aquest tipus de replicació les dades s'actualitzen en el servidor un cop ja han estat modificades en el client. El sistemes amb aquest tipus de rèplica poden experimentar una desincronització que va des de l'ordre de minuts a dies, depenent de la magnitud de la base de dades a replicar.

Aquest tipus de rèplica no seria eficient en el cas que Varpe estigués interessat en que les dades que conté el servidor de Varpe estiguessin sempre actualitzades. D'aquesta manera, si succeeix una incidència en una màquina els operaris de Varpe poden tenir la seguretat que les dades que visualitzen de la màquina corresponent són les actuals.

Dins d'aquests mètodes existeixen diferents maneres per tal d'actualitzar la base de dades replicada un cop s'han modificat en el client. Les tècniques més utilitzades són les següents:

- **Triggers:** Un trigger s'executa quan es compleix una condició establida al realitzar una operació de INSERT, DELETE o UPDATE. Aquest mètode permet ser a l'usuari el responsable de crear el codi per tal que el trigger es dispari en el moment adequat. Existeixen dos tipus de triggers, els triggers per fila que s'executen cada vegada que

hi ha un canvi en un registre de la taula i els triggers de seqüència que s'executen una única vegada encara que la condició que el fa disparar s'hagi donat en més d'una ocasió. Aquesta tècnica ofereix a l'usuari els següents inconvenients:

- La gestió i execució d'un trigger redueix de manera significativa les prestacions.
- Els triggers transfereixen les dades quan són modificades, però no tenen cap seguretat que la transacció s'hagi realitzat amb èxit.
- Els triggers no poden planificar-se, s'executen cada vegada que s'actualitza un registre.
- Si es repliquen diferents taules relacionades, la sincronització de les replicues pot portar-se a terme utilitzant diferents mecanismes.
- L'activació dels triggers no poden anular-se fàcilment en el cas que s'hagi d'abortar la transacció.

En el nostre cas tot i ser una bona opció no s'ha escollit aquest mètode degut a que les taules proporcionades per Varpe no es podien modificar, i per tant no es podia codificar el trigger que permetés actualitzar el servidor si es modificava la base de dades del client.

- **Snapshot:** Els snapshots permeten crear còpies de lectura de la base de dades. Una de les avantatges dels snapshots és que permet fer un backup pàgina a pàgina. D'aquesta manera els snapshots permeten reservar la memòria per tal de generar el backup a mesura que les dades de la base de dades són modificades.

3.2.3 Especificacions del servidor

Existeixen diferents maneres d'administrar el contingut del servidor Varpe. Com en els requeriments inicials (vegeu apartat 2.2) Varpe no va especificar com volia gestionar les dades del servidor. En el projecte s'ha gestionat de la manera següent: quan el propietari d'una màquina de rajos X pateix un incidència amb la màquina, aquest executa el mòdul de sincronització per tal d'enviar les dades al servidor de Varpe on seran analitzades pels

operaris de Varpe. Un cop els operaris han solucionat la incidència les dades continuaran en el servidor fins que siguin actualitzades perquè s'hagi produït una nova incidència.

Tot i que aquesta ha estat la manera d'implementar-ho existeixen altres possibilitats:

- Una de les possibilitats és que un cop solucionada la incidència les dades siguin eliminades del servidor, d'aquesta manera s'aconsegueix més velocitat en el mòdul de sincronització ja que crear les dades de nou es més ràpid que comprovar si cadascun dels registres han estat modificats i actualitzar-les. En aquest cas no caldria implementar tècniques de sincronització ja que el servidor només tindria dades en el cas que hi hagués una incidència sense resoldre. Una de les principals avantatges que proporciona aquest mètode és que com poden comprovar en l'apartat de resultats 4.2 és més ràpid crear de nou tots els registres que actualitzar-los. Per contra aquesta possibilitat fa que el client hagi d'executar el mòdul de sincronització cada vegada que pateix una incidència en la màquina.
- Una altra possibilitat seria que cada vegada que un client adquireix una màquina les dades de configuració quedessin guardades en el servidor i que si en algun moment el client modifica les taules de configuració mitjançant una de les maneres d'actualització de dades explicades en l'apartat 3.2.2 actualitzar les dades. D'aquesta manera Varpe sempre disposaria de les dades de les màquines actualitzades. Naturalment el principal inconvenient d'aquesta possibilitat és la pèrdua de capacitat del servidor perquè hauria de guardar un volum de dades molt gran.

3.2.4 Mòdul de sincronització

El principal objectiu d'aquest mòdul implementat en el projecte és que proporciona als clients de Varpe una eina per tal de fer una rèplica de la base de dades que permet configurar la màquina. El mòdul segueix el diagrama de flux que es pot observar a la Figura 3.4. En el diagrama s'observen tres estats en color groc. Aquests estats són les funcionalitats que presenta el mòdul implementat.

- La primera funcionalitat correspon a la creació de les taules; per tal de crear les taules en el servidor el primer que cal tenir present és que en el servidor es guardaran

les bases de dades de totes les màquines distribuïdes. És per aquesta raó que es important que es pugui separar a quina màquina correspon cadascun dels registres guardats. A la Figura 2.2 podem veure les taules que es creen en el servidor.

Per la creació de les taules en el servidor no s'ha pogut fer una rèplica exacta de les taules de les diferents màquines. En el servidor es tindran les dades de totes les màquines per aquesta raó és molt important poder diferenciar a quina màquina correspon cada registre. Per tal de poder fer aquesta distinció s'ha afegit en les taules un camp amb el nom de màquina on es guarda la IP de la màquina corresponent. Per tal d'implementar aquesta funcionalitat s'ha implementat l'algorisme següent:

numerocolumnes \leftarrow número de columnes SELECT * FROM table

j \leftarrow 1

for *i* = 0 to *i* < *numerocolumnes* **do**

nom[*i*] \leftarrow recuperem el nom de la columna(*j*)

tipus[*i*] \leftarrow recuperem el tipus de la columna(*j*)

j \leftarrow *j* + 1

end for

for *i* = 0 to *i* <= *tipus.length* **do**

tipusSQL[*i*] \leftarrow *tipus*[*i*]

end for

j \leftarrow 1

for *i* = 0 to *i* <= *tipusSQL.length* **do**

if <*i*=1> **then**

inserir el camp corresponen a la IP

else

sentenciaSQL \leftarrow concatenar(*nom*[*i*],*tipusSQL*[*j*])

j \leftarrow *j* + 1

end if

end for

crear la sentència final a executar al servidor

executar la sentència (CREATE TABLE)

- La segona funcionalitat permet fer la rèplica de tots els registres de la base de dades de la màquina al servidor. Aquesta rèplica no és directe ja que en el servidor s'ha d'afegir la IP de la màquina d'on provenen els registres. És molt important que els registres inserits estiguin identificats amb la IP de la màquina, en cas contrari Varpe no tindria manera de saber a quina màquina correspon un registre concret. L'algorisme seguit per tal de fer la rèplica de les dades és el següent:

```

numerocolumnes ← número de columnes SELECT * FROM table
j ← 1
for i = 0 to i < numerocolumnes do
    if <i==1> then
        inserir el camp corresponen a la IP
    else
        nom[i] ← recuperem el nom de la columna(j)
        j ← j + 1
    end if
end for

key ← recuperem el nom del camp de la clau primària
camps ← concatenació de tots els noms recuperats
consulta ← SELECT key FROM table
STATE files ← recuperem el número de files resultant de la consulta anterior
j ← 1
while consulta.next() do
    clau[j] ← recuperem el valor de totes les claus primàries
    j ← j + 1
end while

for i = 1 to i <= files do
    consulta ← SELECT * FROM taula WHERE key = clau[i]
    while consulta.next() do

```

```

for  $j = 1$  to  $j \leq \text{columnnes}$  do
    recuperem el tipus de la columna  $j$ 
    recuperem tots els valor
end for

end while

 $\text{comanda} \leftarrow$  muntem la sentència SQL

Abans d'executar-la al server comprovem si ja existeix

if existeix then
    crida a la funció UPDATE
else
    executem la sentència al server
end if

end for

```

- La tercera i última funcionalitat implementada ha estat l'actualització dels registres. Un cop els registres d'una màquina concreta ja han estat replicats en la base de dades del servidor, si es torna a fer una rèplica de les dades es comprovarà registre a registre i camp a camp que realment existeix un canvi. Si realment existeix un canvi actualitzarà el registre. A continuació es troba l'algorisme implementat:

```

 $\text{consulta} \leftarrow$  SELECT * FROM table

 $\text{columnnes} \leftarrow$  número de columnnes del resultat de la consulta

 $\text{files} \leftarrow$  número de les files del resultat de la consulta

for  $j = 1$  to  $j \leq \text{columnnes}$  do
     $\text{camps}[j] \leftarrow$  recuperem el nom dels camps
     $\text{valors}[j] \leftarrow$  recuperem els valors de la base de dades Paradox
end for

 $\text{consulta} \leftarrow$  SELECT * FROM taula WHERE  $\text{camp}[1] = \text{valors}[1]$  AND MaquinaID=IP

for  $j = 1$  to  $j \leq \text{columnnes}$  do

```

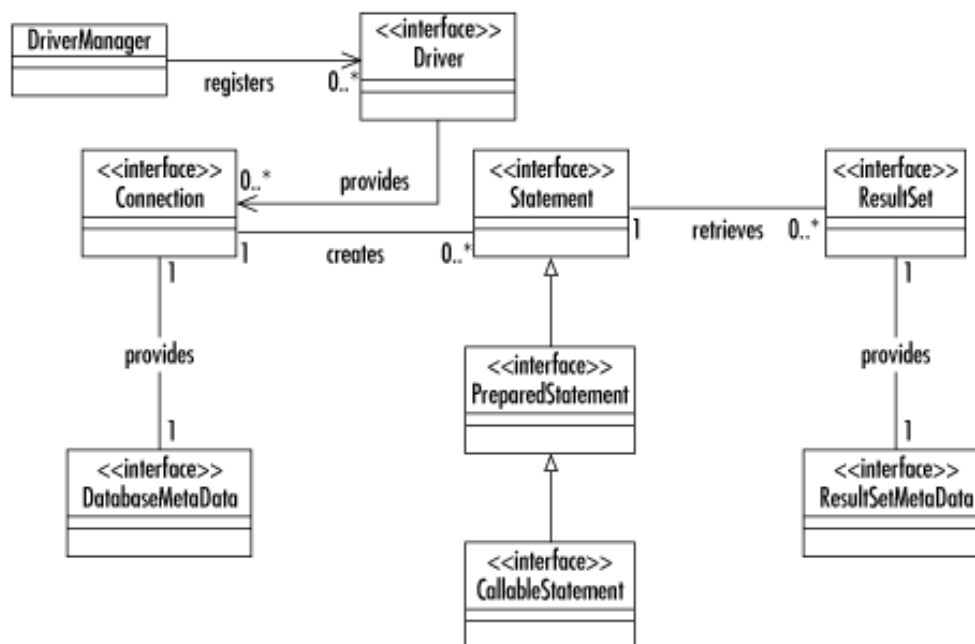


Figura 3.3: Diagrama de les classes de Java

```

    valorsOld[j] ←recuperem els valors del servidor
end for
for j = 1 to j ≤ columnes do
    iguals ←comparem els valors del servidor amb els de la màquina
end for
if iguals then
    print No existeixen diferències
else
    executem la sentència UPDATE table SET valors WHERE valorsOLD
end if
  
```

Per tal de recuperar dades com número de files, número de columnes, etc. Java ens facilita diferents classes útils dins dels paquets `java.sql` i `java.sqlx`. A la Figura 3.3[12] es troben les diferents interfícies útils pel desenvolupament dels algorismes anteriors.

Especificacions per Paradox

Un cop arribats a aquest punt en que ja s'han explicat les diferents tècniques i les diferents eines utilitzades en el desenvolupament en aquesta secció s'explica en detall el mòdul

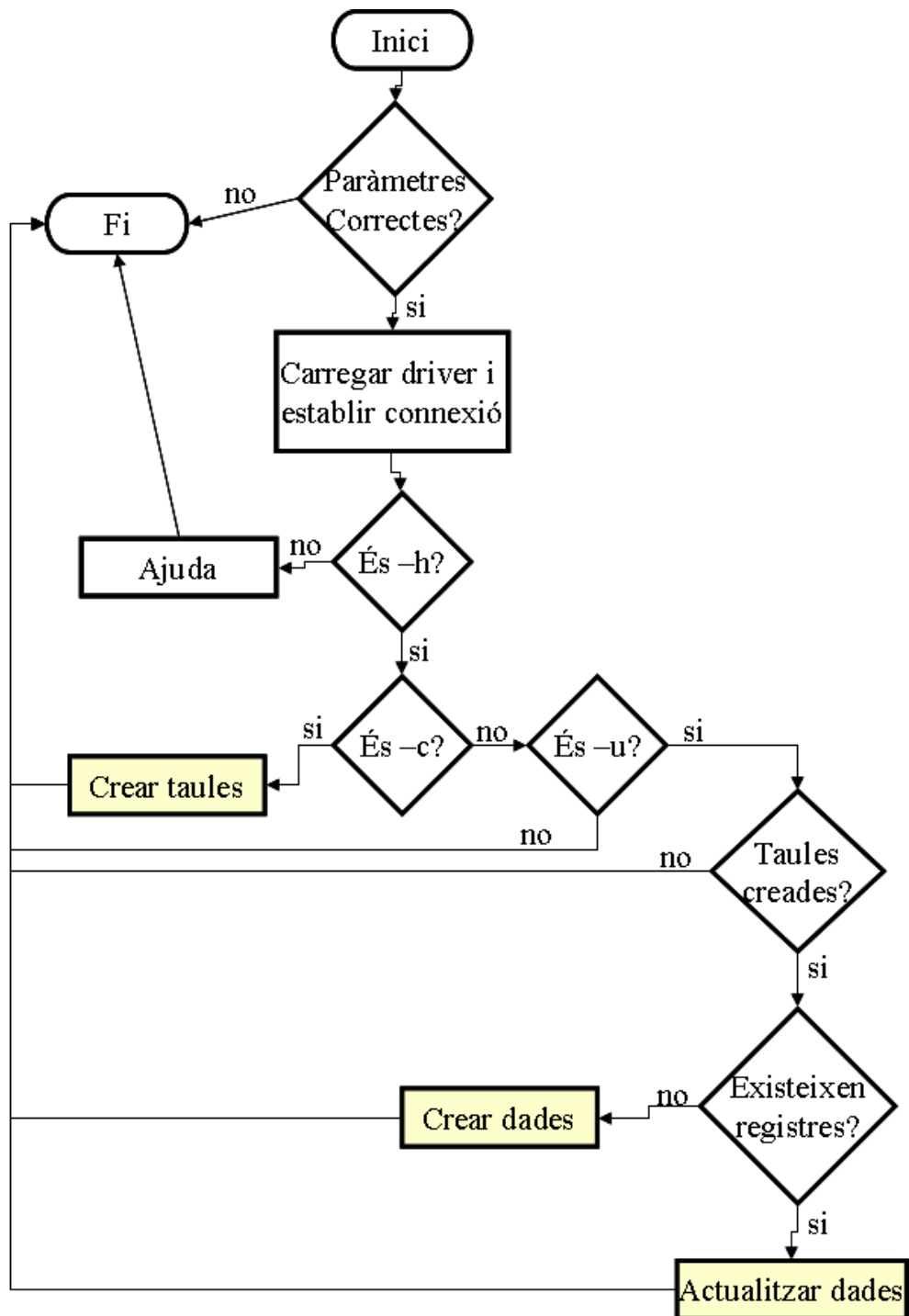


Figura 3.4: Diagrama de flux mòdul sincronització

implementat en el projecte per tal de passar les taules Paradox al servidor MySQL. A la Figura 3.5 es pot veure l'esquema previ que cal tenir per accedir al mòdul de sincronització. Un cop tenim aquesta estructura muntada de la manera explicada en el punt 2.4.3 es procedeix a la implementació del mòdul.

El primer pas és carregar el driver necessari i fer la connexió a la base de dades. D'aquesta manera es poden executar comandes SQL i tenir accés a la base de dades Paradox. Per tal de carregar el driver i establir una connexió amb la base de dades cal fer

`Class.forName("sun.jdbc.odbc.JdbcOdbcDriver")` per carregar el driver JDBC-ODBC i

`Class.forName("com.mysql.jdbc.Driver")` per carregar el driver JDBC-MySQL. Un cop tenim carregats els drivers necessaris cal fer la connexió mitjançant el codi següent:

`url = "jdbc:mysql://" + serverName + "/" + mydatabase`, on `serverName` és la IP del servidor on volem fer la rèplica de les dades y `mydatabase` el nom del DNS associat a la base de dades.

`connexioServer = DriverManager.getConnection(url, username, password)`, amb aquesta instrucció es fa la connexió al servidor de Varpe. Les variables `username` i `password` corresponen al usuari i password creat en el servidor MySQL per tal de tenir accés a la base de dades de Varpe.

Un cop s'ha establert la connexió amb el servidor on es guardarà el resultat de les diferents consultes cal fer la connexió amb la base de dades Paradox. En aquest cas la connexió serà diferent a l'anterior ja que s'ha de carregar el controlador JDBC-ODBC. La forma de fer-ho és la següent:

`connexió = DriverManager.getConnection("jdbc:odbc:varpe","","")`, en el cas concret del projecte la connexió es fa amb `varpe` perquè recordem (vegeu Figura 2.9) que havíem definit amb aquest nom el DNS. Els altres dos paràmetres corresponen al nom de l'usuari i el password corresponent.

Un cop establertes les dues connexions ja es poden executar consultes SQL per tal de seleccionar el registre adequat i guardar-ho al servidor mitjançant també comandes SQL.

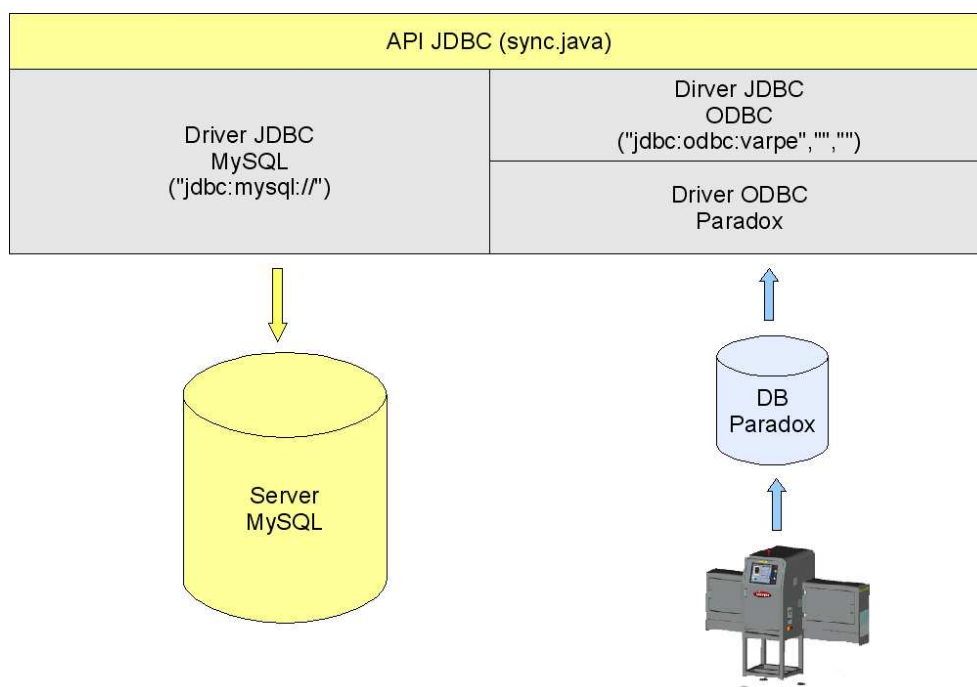
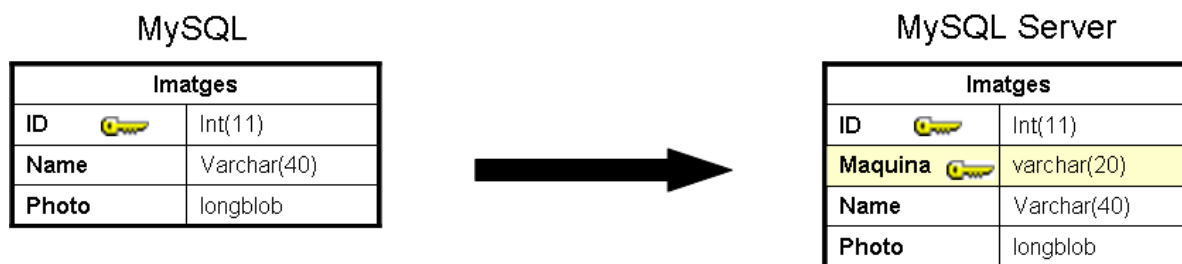


Figura 3.5: *Esquema estructura necessària pel mòdul.*

Especificacions per MySQL

Al desenvolupar el mòdul per Paradox van sorgir problemes associats al controlador ODBC-Paradox proporcionat per Microsoft. La base de dades proporcionada per Varpe al inici del projecte estava en una versió de Paradox que no era compatible amb el controlador ODBC. Com a conseqüència es va proposar a Varpe fer una migració a MySQL o actualitzar la versió de Paradox per tal d'utilitzar el controlador, es va decidir migrar a MySQL. Aquesta migració fa que ja no sigui necessari la configuració del controlador ODBC-JDBC, ja que es pot utilitzar el mateix tipus de connexió que el utilitzat pel servidor MySQL.

La migració a permès també reduir la complexitat del mòdul de sincronització ja que poden evitar algunes accions com el canvi de tipus necessaris per les taules Paradox.

Figura 3.6: Taula *imatges*

3.3 Transmissió d'imatges

Recordem que un dels requeriments de l'empresa Varpe era que volien que el client també disposés d'una eina per tal d'enviar les imatges dels productes contaminats a Varpe, per tal que els operaris de Varpe les poguessin visualitzar i analitzar mitjançant l'aplicació web dissenyada.

Les imatges generades per la màquina de rajos X corresponen a la imatge reconstruïda del producte examinat. Aquestes imatges generades per la màquina tenen un tamany aproximat d'uns 500 kbytes. Respecte a la qualitat de la imatge no és molt important ja que el que realment interessa es veure el contaminant. A la Figura ?? es pot veure una imatge generada per la màquina.

Per tal de donar una solució i aprofitant que estàvem fent la rèplica de la base de dades es va crear dins la mateixa base de dades de la màquina una taula on anar guardant els binaris de les imatges generades. D'aquesta manera aprofitem el mateix mòdul de sincronització explicat en el punt 3.2.4 per enviar les imatges al servidor. La taula creada és la representada a la Figura 3.7:

En el cas de les imatges un cop la taula ha estat replicada en el servidor són eliminades de la base de dades del client. D'aquesta manera s'aconsegueix evitar problemes alhora d'actualitzar el servidor.

3.4 Aplicació Web

L'aplicació web va ser dissenyada per tal de facilitar la visualització de les dades emmagatzemades en el servidor, corresponents a les diferents màquines de Rajos X. Mitjançant

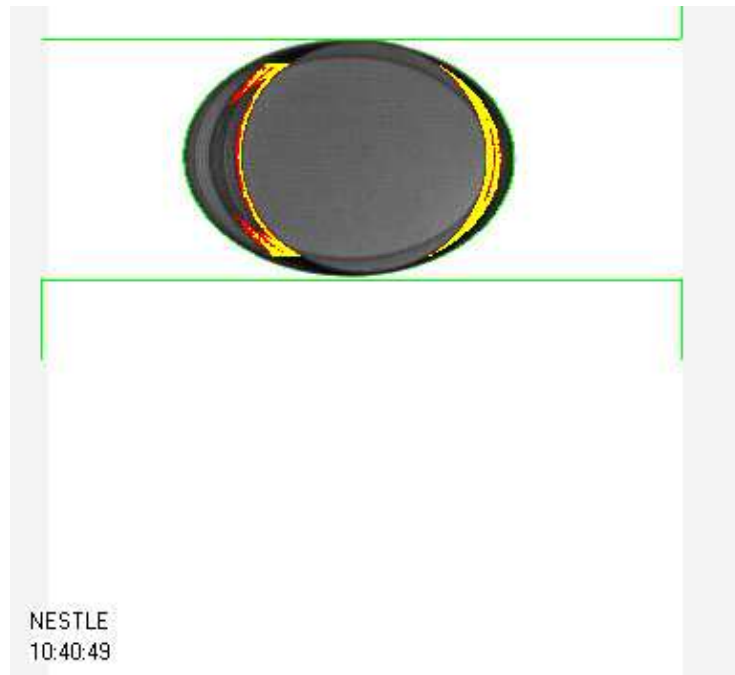


Figura 3.7: *Imatge generada per la màquina Rajos X*

l'aplicació web l'operador pot visualitzar les dades que configuren les diferents màquines via web. Actualment això és impossible ja que les màquines només tenen les dades emmagatzemades de manera local.

Per implementar l'aplicació web s'han implementat diferents pàgines JSP per tal de fer més fàcil la integració amb la web actual de Varpe. Les pàgines creades mostren la informació continguda en el servidor de Varpe, per tal de recuperar aquesta informació la API de Java proporciona les classes de la Figura 3.3 que permeten recuperar les dades mitjançant comandes SQL. Per tal de recuperar les dades cal establir una connexió amb el servidor de Varpe, el codi implementat és el següent:

```
public Connection getBDConnectionSQL()  
{  
    Connection con = null;  
    /* Carreguem el driver JDBC-ODBC */  
    Class.forName("com.mysql.jdbc.Driver");  
  
    String serverName = "192.168.19.130";  
    String mydatabase = "varpe";
```

```
String url = "jdbc:mysql://" + serverName + "/" + mydatabase;
String username = "user_varpe";
String password = "V1rp2";
con = DriverManager.getConnection(url, username, password);
    return con;
}
```

D'aquesta manera establim la connexió amb el servidor i ja es poden executar les comandes SQL pertinents per tal de recuperar la informació desitjada. La manera d'executar la comanda SQL un cop s'ha establert la connexió és la següent:

```
public ResultSet getResultSetSQL( String sqlQuery )
{
    //sqlQuery és la comanda SQL a executar
    ResultSet rs = null;
    /* Creem el Statement i executem la sentència SQL */
    Statement stat = getBDConnectionSQL().createStatement(
        ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);
    rs = stat.executeQuery(sqlQuery);

    return rs;
}
```

Un cop definides les funcions anteriors només cal recuperar les dades necessàries i crear les pàgines seguint l'estil proporcionat per Varpe.

A la Figura 3.8 es troba el diagrama de flux de l'aplicació desenvolupada. La part vermella correspon a la part de l'aplicació desenvolupada per Varpe i la part blava la desenvolupada en el projecte.

Per tal de crear l'aplicació web s'ha seguit l'estil de l'aplicació web amb la qual Varpe gestiona altres màquines com les de control de pes. A la Figura 3.9 es pot visualitzar la pàgina principal de l'aplicació web desenvolupada.

En aquesta web es poden visualitzar totes les màquines de Rajos X que tenen les seves

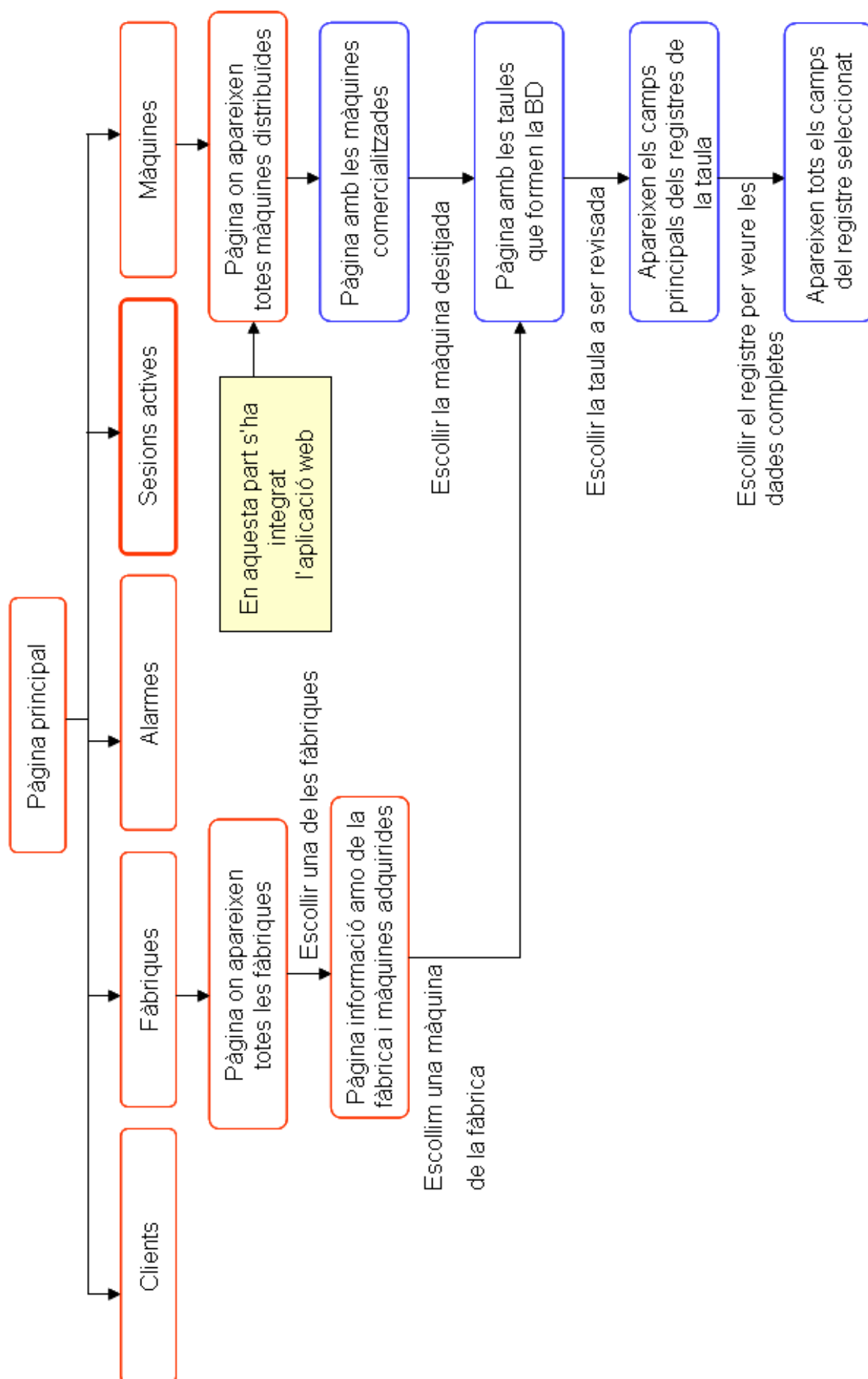


Figura 3.8: Diagrama de flux de l'aplicació web.



Màquina	IP de la màquina
1	192.168.19.131
2	192.168.19.132

Figura 3.9: Pàgina principal de l'aplicació web.




ID taula	Nom de la taula
1	alignment
2	config
3	groupe
4	grouppe
5	imatges
6	interlock
7	log
8	param
9	parame
10	products
11	pwd

Figura 3.10: Pàgina per visualitzar les taules de la màquina.

dades replicades en el servidor MySQL. En el cas de la imatge, tenim dues màquines identificades pel seu número d'IP (*Internet Protocol*). Un cop tenim accés a aquesta pàgina escollim la màquina de la qual volem visualitzar les dades. Per exemple escollim la màquina que té com IP la 192.168.19.132, la pàgina que apareix un cop escollida la màquina és l'observada en la Figura 3.10. La pàgina mostra totes les taules que componen la base de dades de la màquina escollida. Un cop arribat aquest punt l'operador pot escollir la taula de la qual l'interessa visualitzar les dades. Al escollir la taula la pàgina que apareix és la corresponent a la Figura 3.11:

Com que hi ha taules que estan formades per numerosos camps s'ha optat per mostrar els camps més significatius de la taula. Aquests camps són les claus primàries i el tercer



The screenshot shows a web page with a red 'Varpe' logo at the top. Below the logo, the title 'Màquina de Rajos X' is centered. Underneath the title, the word 'pwd' is displayed. A table with three columns: 'Name', 'Maquina', and 'Leve' is shown. The table contains four rows of data.

Name	Maquina	Leve
ABC123	192.168.19.132	2
DEF345	192.168.19.132	1
John	192.168.19.132	4
Richard	192.168.19.132	4

Figura 3.11: Pàgina per visualitzar els registres parcials de la taula seleccionada.

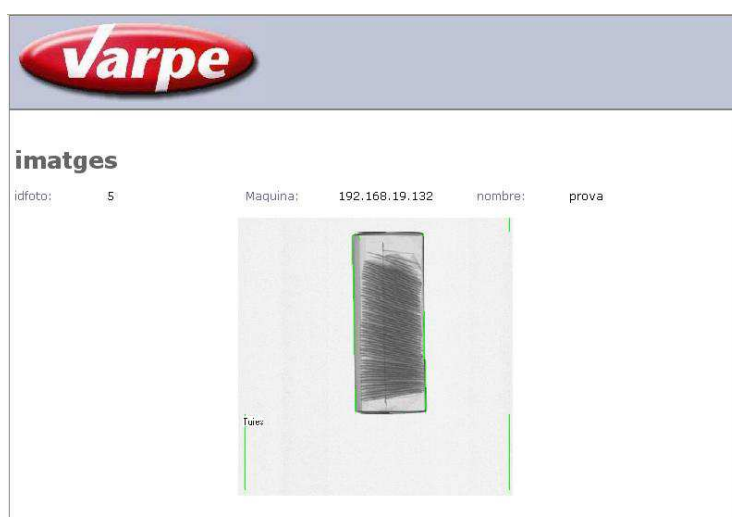


Figura 3.12: Pàgina per visualitzar de manera completa el registre seleccionat.

camp de cada taula. Si volem accedir a la informació completa del registre només cal seleccionar el registre desitjat i apareix una nova pàgina (vegeu Figura 3.12). En aquesta pàgina poden veure totes les dades del registre seleccionat. En el cas mostrat el registre seleccionat pertany a la taula on estan emmagatzemades les imatges generades per la màquina de rajos X, és per aquesta raó que apareix una de les imatges enregistrades per la màquina corresponent a un paquet de pa de motllo.

Capítol 4

Validació de l'aplicació

En aquest capítol es presenten mitjançant taules i gràfiques els resultats obtinguts d'aplicar un joc de proves a l'aplicació desenvolupada en el projecte. D'aquesta manera s'analitza l'eficiència i eficàcia del mòdul de sincronització.

4.1 Joc de proves

Les proves escollides han estat escollides pensant en diferents escenaris en els que es podria trobar el servidor de Varpe.

Per tal de simular diferents màquines distribuïdes en diferents llocs s'ha utilitzat un vmware on es simula el servidor MySQL de Varpe i tres màquines amb IP's diferents. VMware és un sistema de visualització per software que permet simular diferents ordinadors (en el nostre cas màquines de Rajos X) amb unes característiques determinades. D'aquesta manera aconseguim un escenari semblant al de la Figura 4.1.

Per tal de fer la simulació vmware proporciona la interfície de la Figura 4.2 on poden veure els diferents escriptoris dels ordinadors simulats.

El joc de proves escollit per tal de comprovar l'eficàcia i eficiència del mòdul implementat està compost per les següents proves:

- Una màquina executa la comanda de creació de les taules al servidor MySQL. En aquesta prova una màquina executa la comanda que permet crear les taules al servidor sense que el servidor estigui rebent altres peticions.

- Dos o més màquines executen la comanda de creació de taules en el servidor. Tot i que el escenari és altament improbable és fa una prova on dos i tres màquines diferents fan la petició de creació de taules de manera simultània al servidor de Varpe.
- Una màquina executa la comanda que permet fer la rèplica de les dades en el servidor. Un cop creades les taules es comprova quan triga el mòdul en fer la rèplica de la base de dades proposada per Varpe, la qual està composta per uns 1000 registres. La prova s'ha realitzat també amb 2000 i 3000 per tal de poder comprovar l'eficiència amb un volum de dades elevat.
- Una màquina executa la comanda de rèplica tenint guardades les imatges dins de la pròpia base de dades.
- Dos o més màquines executen la comanda per realitzar les seves rèpliques alhora. Com pot donar-se el cas que dos o més clients de Varpe pateixin una incidència en les seves màquines respectives i vulguin replicar les seves bases de dades en els servidor s'ha efectuat una prova on fins a tres màquines diferents enviàvem de manera simultània la petició de rèplica. Igual que la prova anterior aquesta també s'ha executat amb 1000, 2000 i 3000 registres.
- Actualització de les dades. Aquesta opció és necessària per que un cop la màquina ha tingut una incidència les dades no s'esborren del servidor per tant si en algun moment la màquina torna a patir una incidència aquestes dades hauran d'actualitzar-se. Les proves pertinents a l'actualització s'han realitzat executant de manera simultània diferents màquines que intenten reenviar la seva base de dades per tal d'actualitzar-la al servidor.

Com s'ha pogut comprovar mitjançant els diferents resultats mostrats en l'apartat 4.2 l'actualització del servidor no és gaire ràpida, però cal recordar que la implementació del mòdul no permetia modificar les taules proporcionades per Varpe amb la qual cosa per tal de fer una actualització del servidor s'han hagut de comparar tots els registres un per un, d'aquí que els resultats obtinguts no puguin ser més competitius.

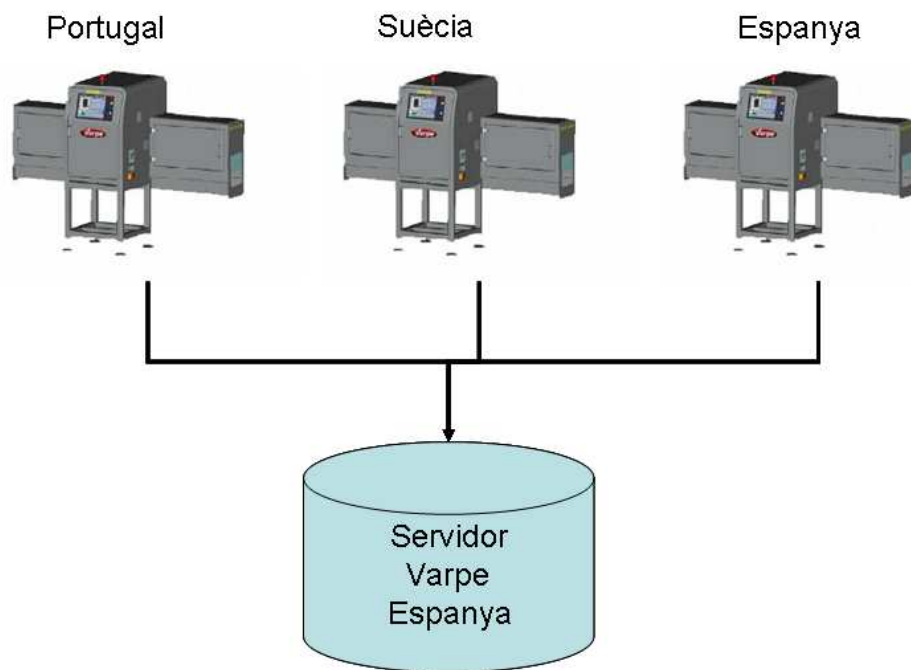


Figura 4.1: Escenari muntat per fer realitzar el joc de proves

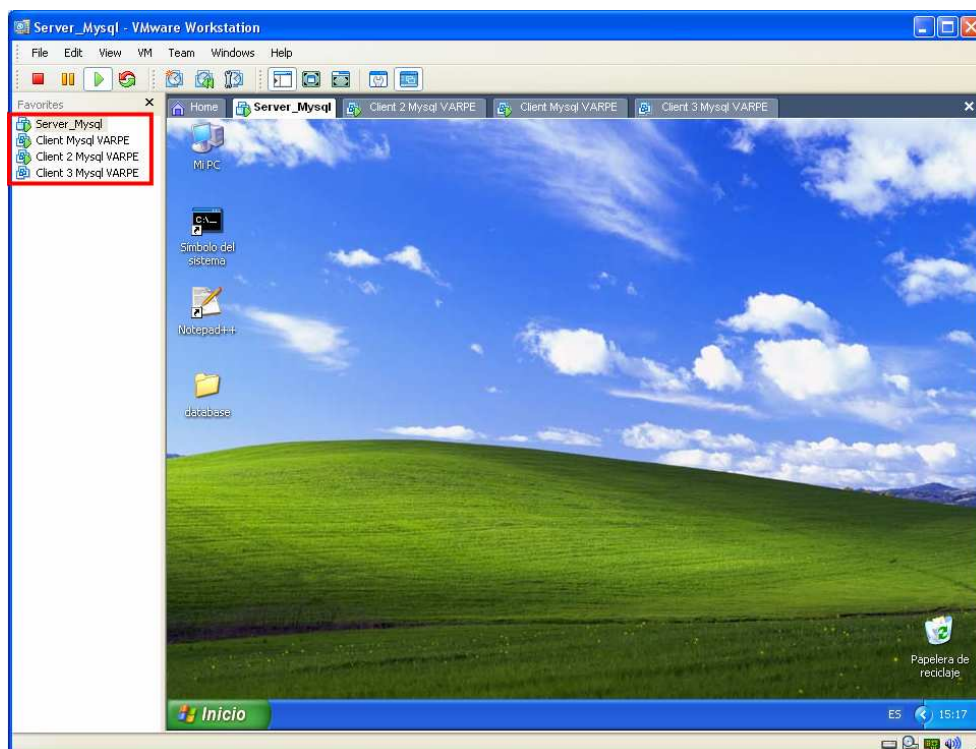


Figura 4.2: Captura del vmware.

4.2 Resultats

Per tal de veure els recursos utilitzats i el temps que triga en executar-se el mòdul de sincronització s'ha implementat una funció. Aquesta funció permet tenir una aproximació del temps i recursos que consumiria l'ordinador associat a una màquina en executar una de les diferents funcions per les quals ha estat programat el mòdul de sincronització.

A continuació es presenten mitjançant diferents taules els resultats a les proves esmentades en l'apartat 4.1.

El mòdul desenvolupat té dos maneres d'executar-se, *java sync IP-màquina -c* per tal de crear totes les taules existents en la base de dades de la màquina en el servidor i *java sync IP-màquina -u* per fer la rèplica o actualització dels registres al servidor.

L'escenari on s'han executant les proves ha estat un portàtil amb quatre vmware funcionant és per aquesta raó que els resultats obtinguts a les diferents proves són només aproximatius ja que els recursos de l'ordinador es veuen dividits per cinc en algunes de les proves. Per totes les proves realitzades la taula imatges conté 10 imatges d'una mida aproximat de 800Kbytes. Les imatges escollides van ser proporcionades per Varpe per tal de fer proves.

La primera prova correspon a la creació de les taules. Totes les màquines tindran les mateixes taules per tant en el moment que les taules estiguin creades en el servidor si una màquina intenta fer la rèplica de la base de dades i abans crea les taules si aquestes estan creades no s'hauran de tornar a crear. Aquesta prova s'ha executat per comprovar que realment si diferents màquines llancen alhora la comanda de crear les taules realment només una les crea. Els resultats obtinguts són els presentats en la següent taula.

#Màquines	1	2	3
Temps	0.31seg	0.92seg	1.65seg
Recursos	0Mb/4Mb	0Mb/4Mb	0Mb/4Mb

Resultats de crear les taules al servidor

Com es pot comprovar la creació de les taules al servidor és eficient ja que, tot i que el mòdul implementat ha de tornar a reconstruir la comanda SQL per crear les taules

afegint el camp que permetrà identificar la màquina el temps de creació és acceptable. En el pitjors del casos, quan tres màquines intenten crear de manera simultània les taules es triga aproximadament un segon en crear-les i els recursos consumits són mínims.

Els resultats presentats en la següent taula són els obtinguts en fer la rèplica de les bases de dades en una, dos i tres màquines alhora. Per tal de comprovar la eficiència i eficàcia del mòdul desenvolupat en el projecte, aquesta prova s'ha fet amb 1100 registres que són els que ens van proporcionar Varpe, amb 2000 i amb 3000 registres que voldria dir que la màquina té una llarga llibreria de productes, per exemple.

#Màquines	1			2					
#Registres	1100	2000	3000	1100		2000		3000	
Màquina	1a	1a	1a	1a	2ª	1a	2a	1a	2a
Temps	6.48seg	17.25seg	32.68seg	8.60seg	8.48seg	31.77seg	34.06seg	36.51seg	41.32seg
Recursos	1Mb/4Mb	3Mb/4Mb	3Mb/4Mb	1Mb/4Mb	1Mb/4Mb	2Mb/4Mb	2Mb/4Mb	3Mb/4Mb	3Mb/4Mb

#Màquines	3								
#Registres	1100			2000			3000		
Màquina	1a	2a	3a	1a	2a	3a	1a	2a	3a
Temps	10.5seg	12.34seg	10.14seg	35.12seg	31.87seg	53.20seg	58.17seg	56.01seg	89.26seg
Recursos	1Mb/4Mb	1Mb/4Mb	1Mb/4Mb	2Mb/4Mb	2Mb/4Mb	7Mb/10Mb	2Mb/4Mb	4Mb/4Mb	2Mb/4Mb

Resultats en fer la rèplica de la base de dades al servidor

La rèplica de les dades és ràpida tot i que el mòdul ha d'afegir un camp a cadascun dels registres que replica per tal d'identificar de la màquina que provenen. A la taula podem veure que si tres màquines diferents intenten fer la rèplica de les seves dades (en el pitjor dels casos 3000 registres) alhora triguen un minut aproximadament.

Per tal de comprovar si el fet d'enviar les imatges a través de la base de dades afecten al rendiment de l'aplicació es va realitzar una prova on es replica la mateixa base de dades anterior però eliminant la taula imatges. Els resultats de la taula següent determinen que enviar les imatges a través de la base de dades és factible perquè la diferència de temps és de 10 segons aproximadament.

#Màquines	2	
#Registres	2000	
Màquina	1a	2a
Temps	24.35seg	25.40seg
Recursos	2Mb/4Mb	2Mb/4Mb

Resultats en fer la rèplica de la base de dades al servidor sense la taula imatges

Les últimes proves realitzades corresponen a l'actualització de les dades en el servidor. Els resultats de la taula següent permeten veure que la tècnica utilitzada per fer l'actualització de les dades no és la adequada. Al no poder utilitzar tècniques com els triggers explicades en la secció 3.2.2 l'actualització de les dades al servidor són molt costoses ja que l'actualització s'ha de fer registre a registre tot comparant cadascun dels camps. Aquests resultats es podrien millorar si Varpe modifiqués les taules afegint *triggers* que es desapareixin en el moment en que s'actualitza algun registre.

#Màquines	1			2					
#Registres	1100	2000	3000	1100		2000		3000	
Màquina	1a	1a	1a	1a	2ª	1a	2a	1a	2a
Temps	28.78seg	12.02m	26.57m	41.82seg	42.79seg	19.80m	19.63m	43.45m	43.43m
Recursos	1Mb/4Mb	1Mb/4Mb	2Mb/4Mb	2Mb/4Mb	2Mb/4Mb	2Mb/4Mb	4Mb/4Mb	3Mb/4Mb	4Mb/4Mb

#Màquines	3								
#Registres	1100			2000			3000		
Màquina	1a	2a	3a	1a	2a	3a	1a	2a	3a
Temps	56.12seg	1.05m	56.42seg	29.47m	29.47m	29.43m	60.75m	60.65m	60.98m
Recursos	2Mb/4Mb	2Mb/4Mb	2Mb/4Mb	1Mb/4Mb	1Mb/4Mb	2Mb/4Mb	3Mb/4Mb	4Mb/4Mb	2Mb/4Mb

Resultats en fer l'actualització de la base de dades al servidor

A la figura 4.3 podem veure com el temps emprat per fer la rèplica de les dades és linialment proporcional al número de registres replicats. A mesura que el número de registres augmenta el temps que triga en fer la rèplica també augmenta. Si comparem les tres gràfiques en el punt que corresponen a 1100 registres que són els proporcionats per Varpe es pot observar com la diferència entre les tres és aproximadament de 10 segons. Aquesta diferència podrien ser conseqüència de la divisió de recursos del ordinador utilitzat per fer la simulació.

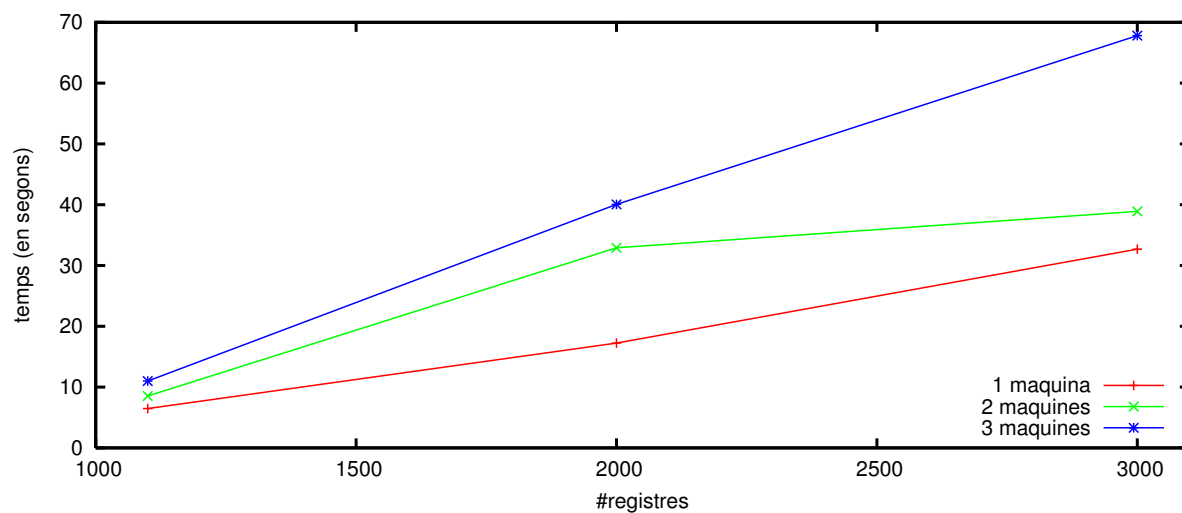


Figura 4.3: Gràfica resultant de fer la rèplica de la base de dades

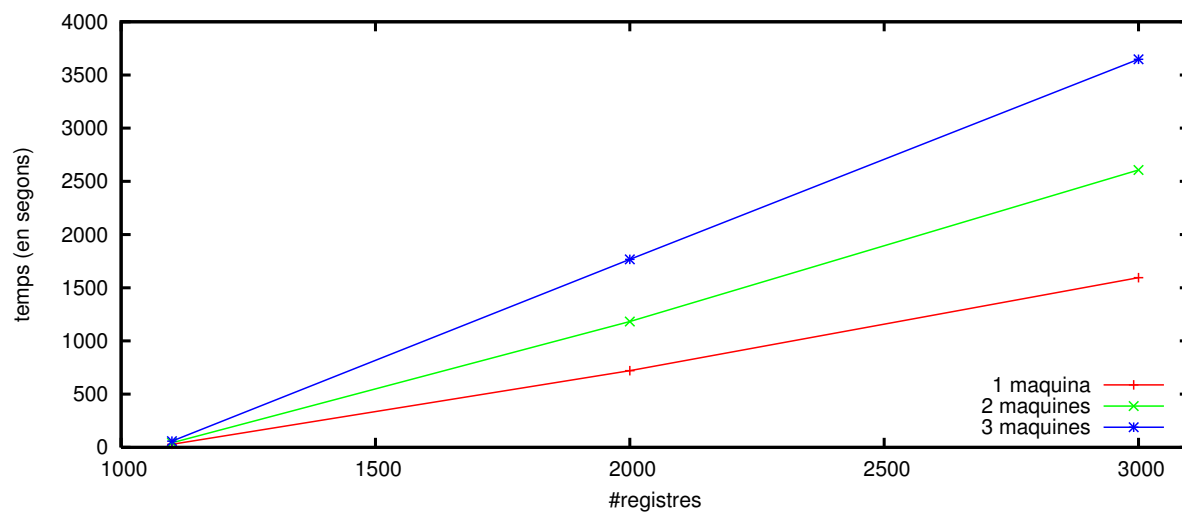


Figura 4.4: Gràfica resultant de fer l'actualització de la base de dades

A la figura 4.4 podem veure la gràfica resultant d'actualitzar les dades al servidor. Aquestes dades poden variar depenent del número de registres que ha d'actualitzar. Aquesta gràfica permet deixar de manera clara que replicar la base de dades és més eficaç que actualitzar-la.

Capítol 5

Planificació

La planificació és una de les parts més importants de tot projecte. En aquest capítol es dóna una visió general de quines han estat les tasques desenvolupades i el temps necessari per fer-les.

5.1 Planificació temporal

En l'actualitat cada vegada és més important abans de tot projecte fer una bona planificació d'aquest. És per aquesta raó que s'han desenvolupat eines tant útils com Microsoft Project on es poden planificar cadascuna de les parts del projecte.

L'objectiu de la planificació d'un projecte de software com el desenvolupat és proporcionar un marc de treball que permeti estimar els costos i recursos necessaris per dur a terme el projecte. Un projecte té diferents fases on s'estableixen els requeriments, objectius, recursos, etc. A la Figura 5.1 es poden veure les diferents parts d'un projecte.

Les parts més important són:

- **Fase de planificació:** En aquesta fase s'estableix com l'equip de treball ha de satisfer les restriccions de prestacions, planificació temporal i cost.
- **Fase d'execució:** Representa el conjunt de tasques que componen el projecte. En aquesta s'especifiquen les característiques tècniques.
- **Fase d'entrega:** Tot projecte té una data límit d'entrega. Aquesta fase és molt

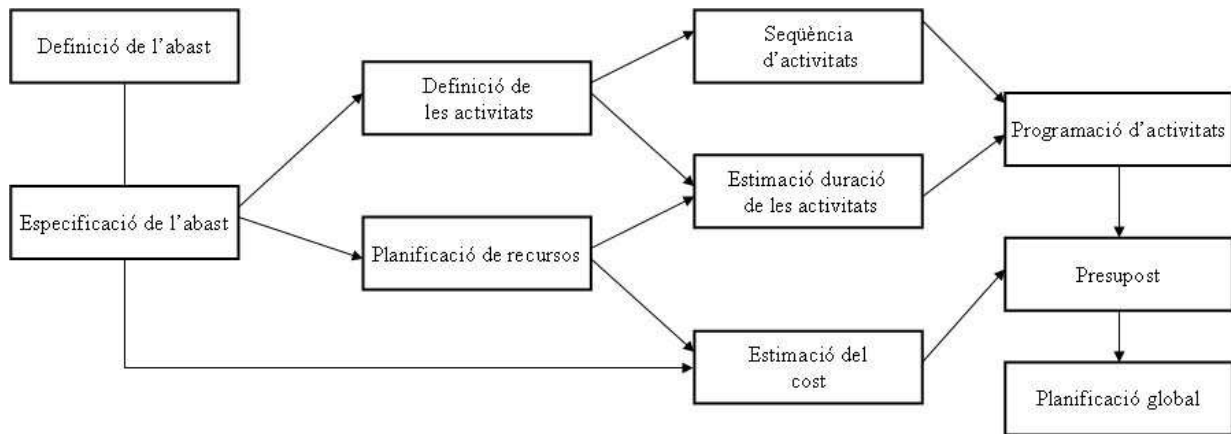
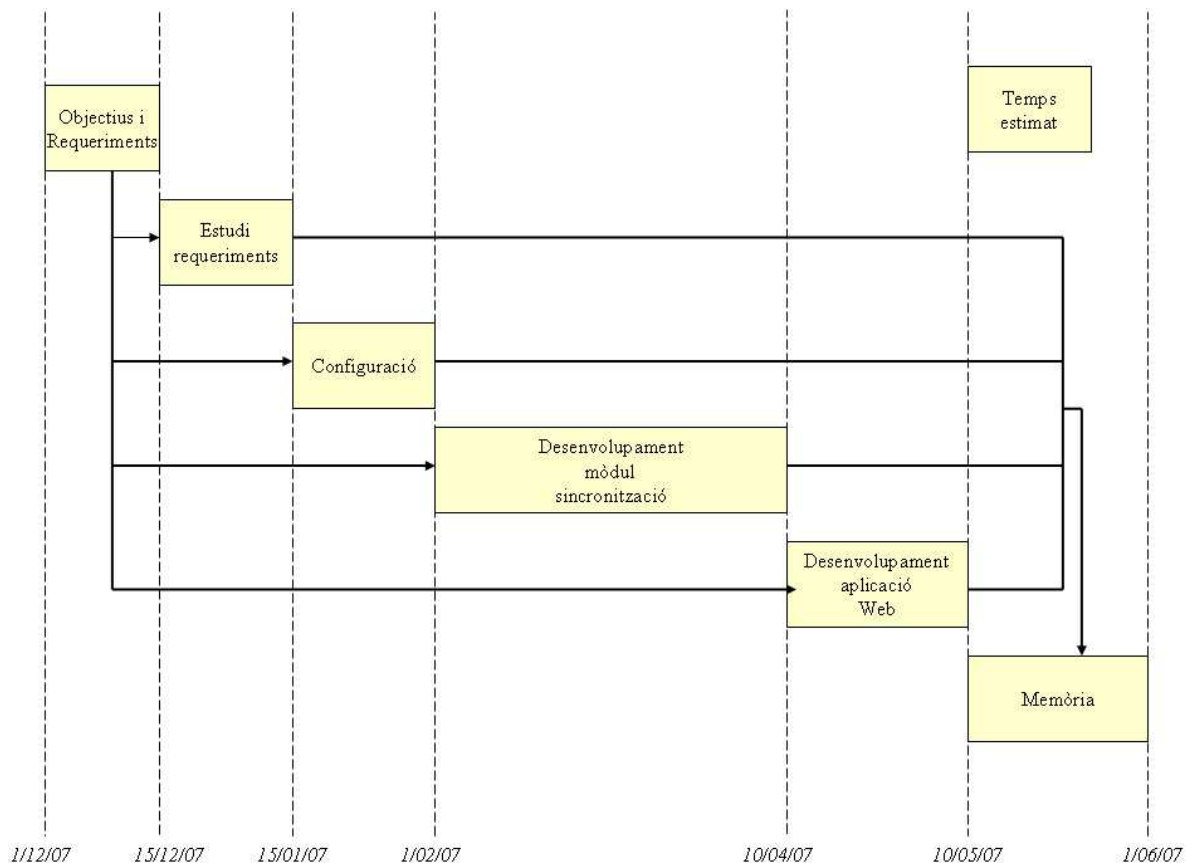


Figura 5.1: *Esquema de les fases d'un projecte*

també és important no només per l'entrega del projecte si no perquè pot provocar costos imprevistos com a conseqüència de la negociació amb el client als inicis del projecte.

En el cas particular del projecte, sistema de control remot d'una màquina de rajos X via web, un cop marcats els requeriments de Varpe es va planificar el temps per tal d'assolir els objectius marcats (vegeu apartat 1.2) en el temps del que es disposava per elaborar el projecte. La planificació presentada es va tenir que modificar per diferents motius i problemes associats al projecte. La planificació general efectuada va ser la següent:

- Un primer mes per tal que Varpe ens proporcionés les dades necessàries i per fer una primer estudi del objectius ha assolir per satisfer els requeriments de Varpe.
- Mig mes per configurar l'ordinador de manera que poguéssim disposar de tots els recursos necessaris pel desenvolupament del projecte. Intallació del diferents software utilitzat, NetBeans, Vmware, Controladors, etc.
- Un cop instal·lat tot el software necessari es va planificar un més i mig per desenvolupar el mòdul de sincronització. Aquest temps es va allargar com a conseqüència de la migració de Paradox a MySQL.
- Per últim, un mes més per desenvolupar l'aplicació web.
- L'últim mes es reserva per l'elaboració de la memòria.

Figura 5.2: *Planificació inicial del projecte*

Per realitzar aquesta planificació es va tenir present que el temps dedicat a l'elaboració del projecte eren de 2 a 4 hores diàries.

A la Figura 5.3 es pot veure de manera esquemàtica la planificació esmentada amb anterioritat.

5.2 Anàlisi de costos

Per avaluar els costos associats al desenvolupament del projecte s'ha establert un preu/hora per l'enginyer desenvolupador. Aquest preu és 9 €/hora, el preu ha sigut establert tenint en compte el que cobra un enginyer informàtic en finalitzar els estudis.

A continuació es presenten de manera detallada les hores dedicades a cada punt del projecte.

Com es pot veure a la taula 5.1 les hores totals destinades al desenvolupament del projecte han estat aproximadament d'unes 156 hores per tant el cost total de l'aplicació

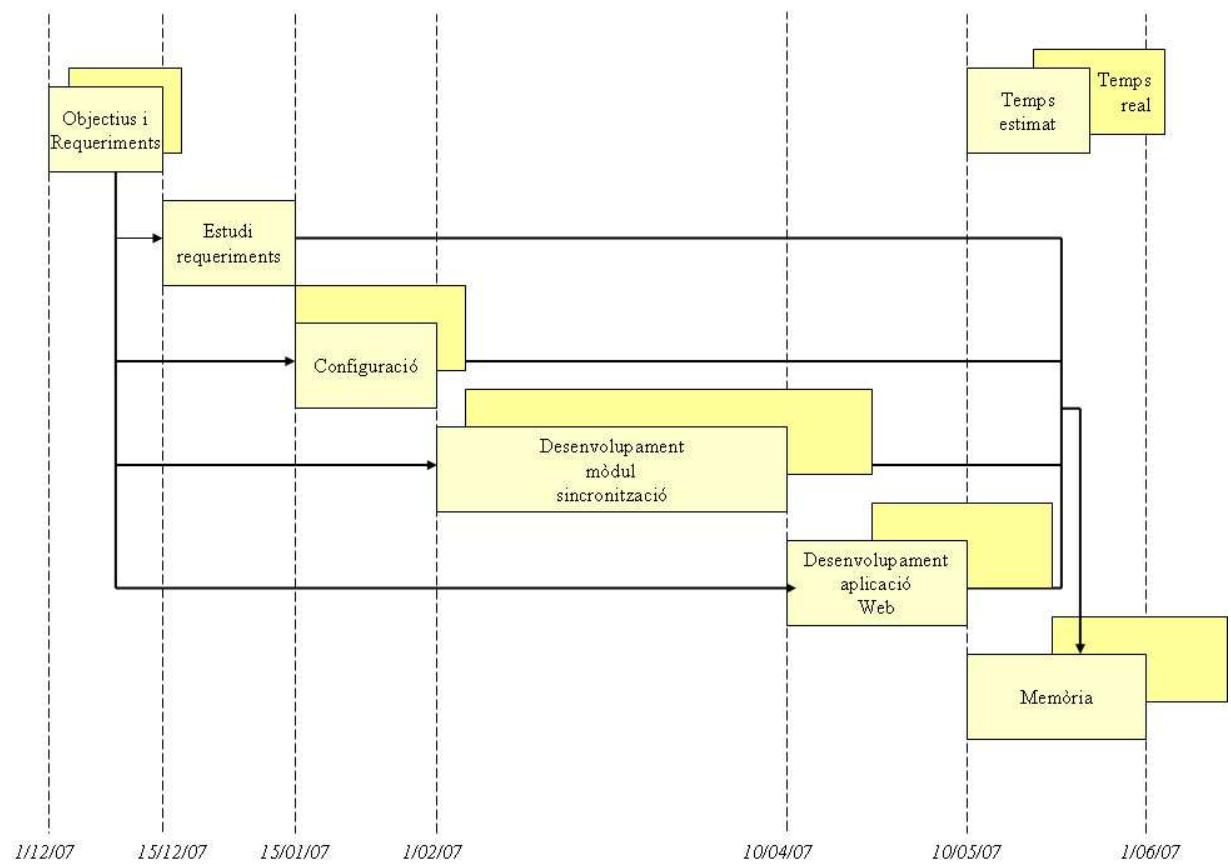


Figura 5.3: *Planificació final del projecte*

seria 1431 €.

Taula 5.1: *Especificacions de les hores per tasca.*

TASCA	TEMPS
Objectius i Requeriments	
Estudi objectius	5
Estudi requeriments	3
Reunió amb el director	3
TOTAL	11
Configuració ordinador treball	
Estudi tipus de controladors disponibles	4
Instal·lació controladors	4
Instal·lació programes	4
TOTAL	12
Mòdul sincronització	
Estudi diferents maneres de replicar bases de dades	20
Desenvolupament mòdul sincronització per Paradox	30
Reunió amb el director	2
Desenvolupament mòdul sincronització per MySQL	10
TOTAL	62
Transmissió imatges	
Estudi opcions transmissió imatges	2
Implementació transmissió imatges	5
Proves	7
Reunió de validació amb el director	2
TOTAL	16
Aplicació web	
Desenvolupament aplicació web	20
Proves	8
Reunió de validació amb el director	2
TOTAL	30
Memòria	
Redacció memòria	20
Revisió directr	3
Correccions de la memòria	5
TOTAL	28
TOTAL GENERAL	159

Capítol 6

Conclusions

6.1 Resum

El projecte titulat Sistema de control remot d'una màquina de Rajos X via web ha permet ha l'empresa Varpe solucionar una sèrie de problemes que tenia fins ara amb la administració de les màquines de rajos X distribuïdes a moltes empreses.

El projecte ha assolit els objectius marcats en l'apartat 2. Mitjançant la rèplica de les dades s'ha aconseguit tenir totes les bases de dades que configuren les diferents màquines dins del mateix servidor. Aquest pas permet als operaris de Varpe poder resoldre les incidències de les màquines sense la necessitat de desplaçar-se fins on es troba la màquina per tal de comprovar que els paràmetres de configuració són els correctes. En la creació del mòdul van sorgir diferents problemes amb els controlador, per aquesta raó tot i no estar definit en els objectius inicials s'ha fet una migració a MySQL.

Un altre requeriment era la necessitat de poder enviar al servidor de Varpe les imatges generades per la màquina. Les imatges permeten a l'usuari de la màquina determinar si el producte examinat està contaminat o no. Si els paràmetres de la màquina no estan ben configurats poden donar-se falsos positius és per aquesta raó que els operaris volen en el servidor les imatges generades per la màquina. Visualitzant les imatges es pot comprovar si el producte realment està contaminat o la màquina està mal configurada i per aquesta raó ha determinat que el producte està contaminat sense estar-ho. Per tal de solucionar aquest requeriment s'ha implementat un nova taula, aprofitant la migració a MySQL que

està efectuant Varpe, per tal de guardar els binaris de les imatges generades per la màquina de rajos X. D'aquesta manera quan s'executa el mòdul de sincronització també s'envien les imatges cap al servidor.

Per tal de visualitzar les dades de manera agradable s'ha integrat a l'aplicació web de Varpe una sèrie de pàgines web que permeten visualitzar les diferents màquines que han executant el mòdul que permet fer la rèplica. Un cop escollida la màquina desitjada es poden visualitzar els registres concrets de totes les taules replicades. Mitjançant l'aplicació web també es poden visualitzar les imatges generades per la màquina.

Així doncs, el projecte ha assolit els tres requeriments principals proposats per l'empresa Varpe per l'elaboració del projecte.

6.2 Futures millores

- La primera millora que es proposa proporcionarà una millora de rendiment en l'aplicació. Aquesta millora seria proporcionada per la utilització de triggers en les bases de dades. Si s'afegeix un trigger en la base de dades proporcionada pel client de manera que aquest es dispari cada vegada que hi ha un canvi a la base de dades, l'actualització de les dades en el servidor seria més efectiva ja que no caldria emportar-se tot el registre per comprovar si aquest ja està al servidor MySQL o no. Els triggers ens permetrien tenir el servidor actualitzat en tot moment. D'aquesta manera Varpe podria solucionar en temps real la incidència d'una màquina ja que no li caldria que el client executés el mòdul de rèplica ja que les dades estarien actualitzades.
- La segona millora fa referència a la gestió de les dades via web. Com que les màquines poden arribar a generar molta informació, per exemple la taula log, es podria fer que l'operari de Varpe pogués eliminar via web aquells registres de les taules que no li aporten cap tipus d'informació necessària per resoldre la incidència. D'aquesta manera també es controlaria la grandària de la base de dades del servidor.
- Alhora de gestionar les dades una altra possibilitat seria eliminar les dades de la màquina una vegada han estat replicades per tal de que els operaris de Varpe resolguin

la incidència. D'aquesta manera un cop canviats els paràmetres pertinents pel bon funcionament de la màquina la base de dades tornaria a ser replicada en el client.

- Aprofitant que actualment Varpe està fent una migració de la base de dades Paradox a MySQL es podria modificar les taules Param i unificar-les, ja que com hem dit aquestes taules guarden la mateixa informació el que canvia són l'idioma dels camps de tipus Varchar.
- Com s'ha pogut comprovar en les proves realitzades en l'apartat 4.2 és més ràpid crear de nou les dades en el server que actualitzar les dades que aquest ja tenia. És per aquesta raó que una de les futures millores seria eliminar les dades del servidor MySQL de Varpe un cop la incidència de la màquina ha estat solucionada. D'aquesta manera en la base de daes del server només s'emmagatzemarien les dades de les màquines amb incidències pendents.

Bibliografia

[1] **Sun Microsystems.**

URL: <http://java.sun.com/j2se/1.3/docs/api/>. Maig 2007.

[2] **Manual Paginació en JSP.**

URL: <http://www.mononeurona.org/archivos/jsp.pdf>. Maig 2007.

[3] **Sun Microsystems.**

URL: <http://java.sun.com/javase/downloads/index.jsp>. Desembre 2006.

[4] **Tutorial JDBC.**

URL: <http://programacion.com/java/tutorial/jdbc/>. Febrer 2007.

[5] **Manual MySQL.**

URL: <http://dev.mysql.com/doc/refman/5.0/es/index.html>. Maig 2007.

[6] **Connector JDBC-MySQL**

URL: <http://www.mysql.com/products/connector/j/index.html>. Gener 2007.

[7] **Tutorial Latex**

URL: <http://frodo.elon.edu/tutorial/tutorial/>. Maig 2007.

[8] **Tutorial JSP**

URL: <http://tejo.usal.es/fgarcia/docencia/poo/02-03/trabajos/S2T3.pdf>. Maig 2007.

[9] **Protocol 2PC**

URL: <http://sinbad.dit.upm.es/docencia/grado/curs0304>. Maig 2007.

[10] **Pàgina oficial Tomcat**

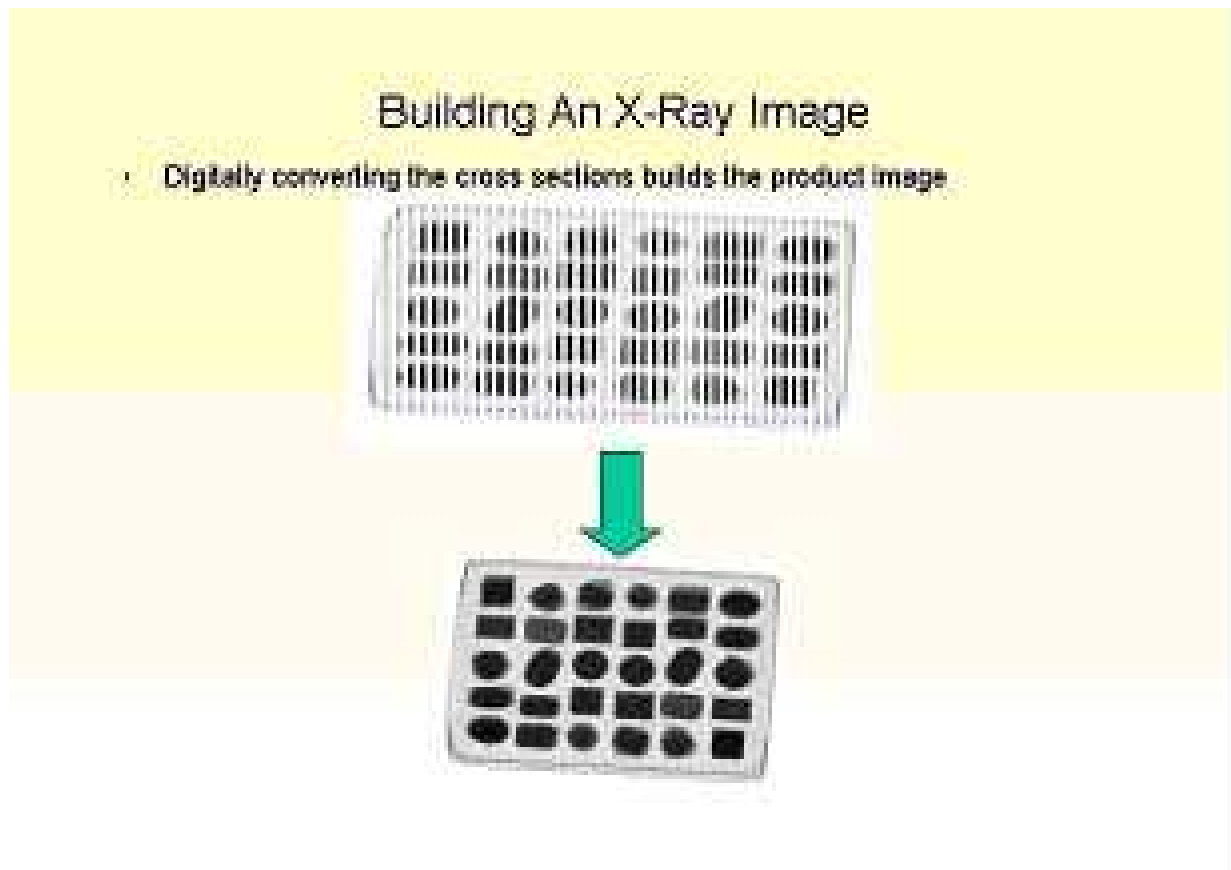
URL: <http://tomcat.apache.org/>. Consulta: Febrer 2007.

- [11] Thomas M Connolly, Carolyn E. Begg, Sistemas de bases de datos, Addison Wesley, 2005.
- [12] George Reese, Database Programming with JDBC and Java, 2nd Edition, 2000.
- [13] Budi Kurniawan, Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions, 2002.

Apèndixs

Apèndix A

Algorisme d'inspecció

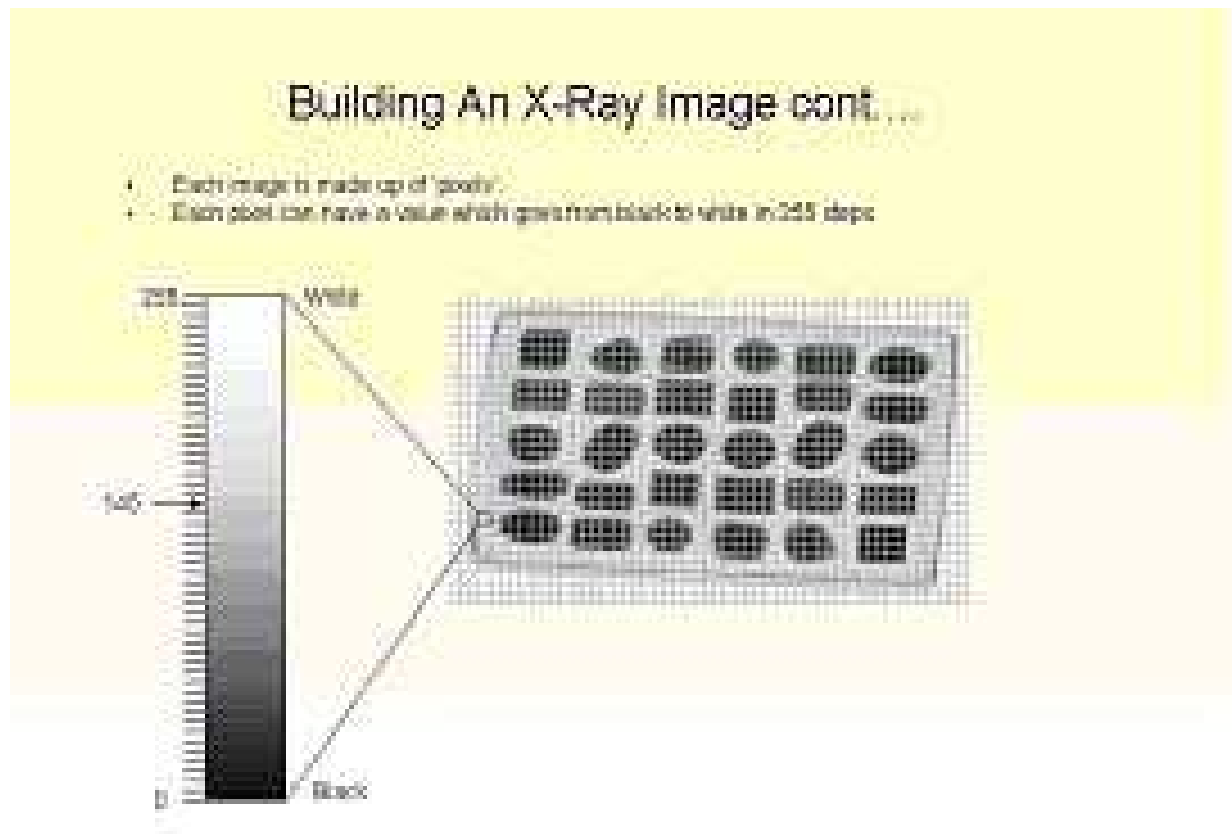


La tarjeta de procesamiento de imágenes recoge los datos línea por línea (como en una imagen de televisión). La frecuencia de captura de las líneas queda establecida por la frecuencia de línea. Una frecuencia de línea de 600 significa que se llevan a cabo 600 exploraciones por segundo.

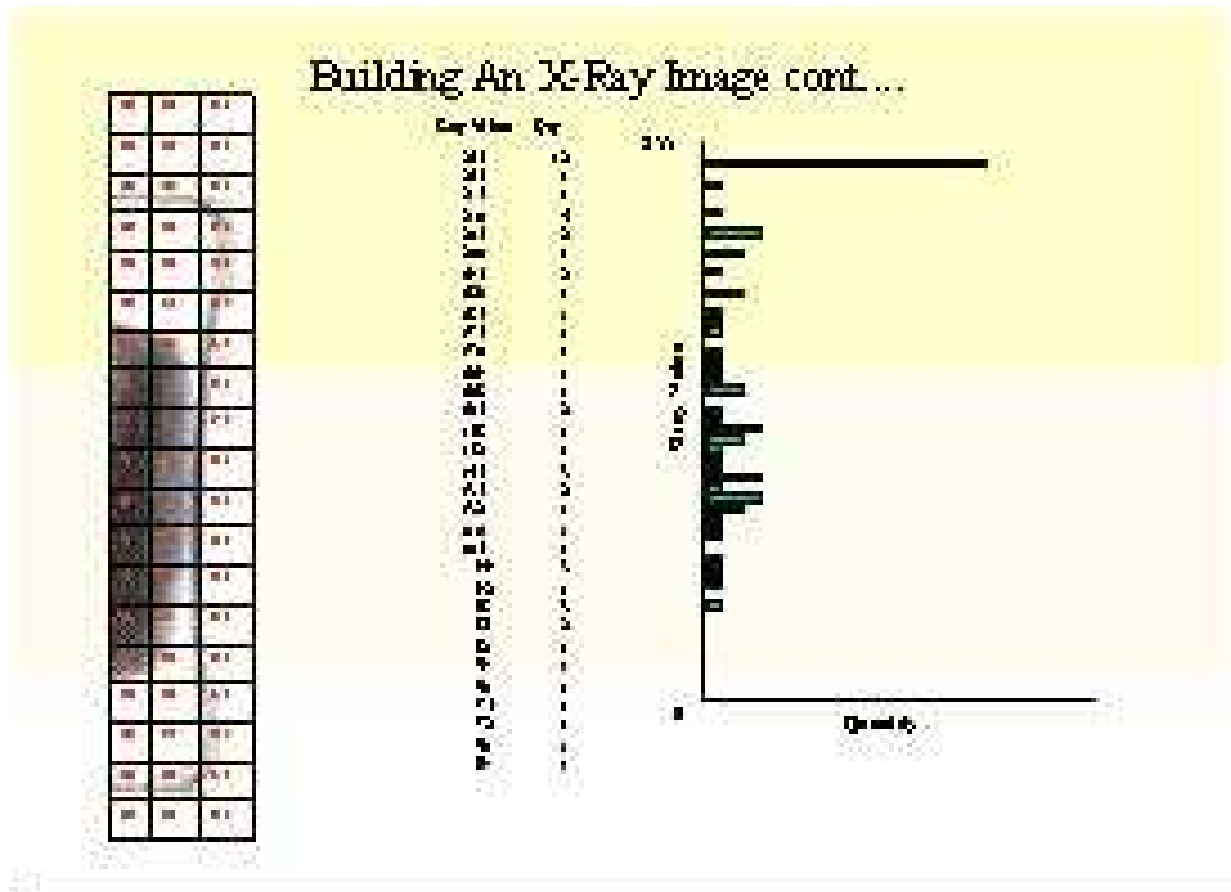


Los ejemplos anteriores muestran el efecto de la frecuencia de línea sobre la imagen y la capacidad de contaminación.

Construcción de un HISTOGRAMA



La imagen queda capturada por la memoria del ordenador. Las líneas de exploración están formadas por píxeles. Cada píxel tiene un valor que va del negro al blanco en 255 graduaciones.



El sistema agrupa cada nivel de valor de gris según el número de píxeles de dicho valor. A ésto se le llama formación de histogramas. Como se muestra más arriba.